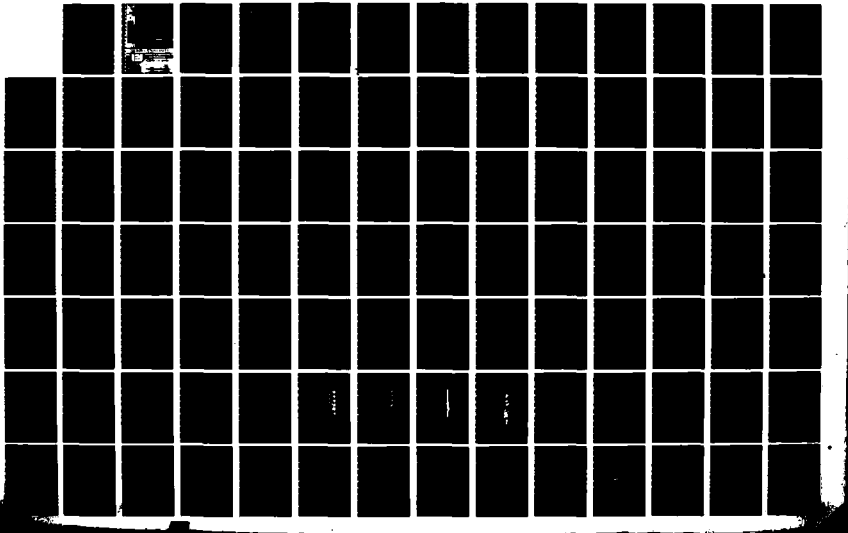
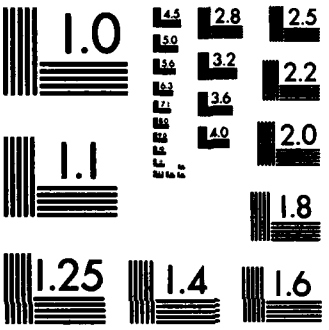
AD-A135 744 ANALYSIS OF THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOWS USING OPTIMUM GRID(U) INDIANA UNIV-PURDUE UNIV AT INDIANAPOLIS SCHOOL OF ENGINEERIN.. A ECER DEC 82 1/2

UNCLASSIFIED AFOSR-TR-83-1052 AFOSR-80-0258 F/G 20/4 NL





AFOSR-TR- 83-1052

Final Report

submitted to

Air Force Office
of Scientific Research
Bolling Air Force Base
Washington DC 20332

for

Analysis of Three-Dimensional Transonic Potential Flows Using Optimum Grid

(Grant No: AFOSR-80-0258)

for the period of

January 1, 1982-December 31, 198

PURDUE UNIVERSITY
School of Engineering and Technology
at Indianapolis
Indiana University—Purdue University at Indianapolis

Final Report

Submitted to

Air Force Office
of Scientific Research
Bolling Air Force Base
Washington DC 20332

for

Analysis of Three-Dimensional Transonic Potential Flows Using Optimum Grid

(Grant No: AFOSR-80-0258)

for the period of

January 1, 1982-December 31, 1982

by

Akin Ecer, Principal Investigator
Division of Engineering
Purdue University at Indianapolis
1201 E. 38th Street
Indianapolis, IN 46205

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83-1052	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ANALYSIS OF THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOWS USING OPTIMUM GRID		5. TYPE OF REPORT & PERIOD COVERED FINAL 1 JAN 82 - 31 DEC 82 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) AKIN ECER		8. CONTRACT OR GRANT NUMBER(s) AFOSR-80-0258
9. PERFORMING ORGANIZATION NAME AND ADDRESS PURDUE UNIVERSITY AT INDIANAPOLIS DIVISION OF ENGINEERING 1201 E 38TH STREET, INDIANAPOLIS, IN 46205		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2307/A1
11. CONTROLLING OFFICE NAME AND ADDRESS AIR FORCE OFFICE OF SCIENTIFIC RESEARCH/NA BOLLING AFB, DC 20332		12. REPORT DATE December 1982 13. NUMBER OF PAGES 105
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release; Distribution Unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

TRANSONIC FLOW
FINITE ELEMENT
BODY FITTED GRID
THREE-DIMENSIONAL FLOW
POTENTIAL FLOW

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A three-dimensional finite element procedure is developed for the analysis of three-dimensional transonic flows and applied to the analysis of wing-body combinations. A finite element grid generation scheme for three-dimensional bodies with complex geometries is presented. The design of efficient, body-fitted computational grids with isoparametric mappings, as well as the application of higher-order finite elements in analyzing transonic potential flows are investigated. Two different computational grids were designed

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

and studied with a numerical scheme based on the density upwinding in the supersonic regions. A pseudo-unsteady type formulation is employed in determining a steady-state solution. It is concluded that the grid generation scheme is quite flexible and efficient for generating solution adaptive grids and providing local refinements in the sensitive flow regions. Also, it is shown that the employed numerical scheme with higher-order elements at flow regions of high gradients produced results which compare favorable with experimental data.

TABLE OF CONTENTS

ABSTRACT

A three-dimensional finite element procedure is developed for the analysis of three-dimensional transonic flows and applied to the analysis of wing-body combinations. A finite element grid generation scheme for three-dimensional bodies with complex geometries is presented. The design of efficient, body-fitted computational grids with isoparametric mappings, as well as the application of higher-order finite elements in analyzing transonic potential flows are investigated.

Two different computational grids were designed and studied with a numerical scheme based on the density upwinding in the supersonic regions. A pseudo-unsteady type formulation is employed in determining a steady-state solution. It is concluded that the grid generation scheme is quite flexible and efficient for generating solution adaptive grids and providing local refinements in the sensitive flow regions. Also, it is shown that the employed numerical scheme with higher-order elements at flow regions of high gradients produced results which compare favorable with experimental data.

LIST OF FIGURES

LIST OF SYMBOLS

a	Local speed of sound.
Cp	Specific heat at constant pressure.
Cv	Specific heat at constant volume.
d	Streamline direction.
e	Element number subscript.
f_j	Flux vector for node i
I,J,K	Natural coordinate system.
$\underline{J}$	Jacobian matrix for volume transformation.
$\lvert\underline{J}\rvert$	Determinant of Jacobian matrix.
$\underline{J}_{xy}$	Jacobian matrix for surface transformation.
$\underline{K}$	Coefficient matrix.
$\underline{K}^o$	Constant coefficient matrix.
M	Local mach number.
M_∞	Free stream mach number.
$\underline{N}$	Shape functions for 3-D elements.
N_i	Shape function for i^{th} node.
$\underline{n}$	Unit outward normal vector.
n	Time step.
P	Local fluid pressure.
p_i	Gauss integration point.
q	Local flow speed.
R	Universal gas constant.
s	Entropy of fluid particals per unit mass.
S	Surface.
t	Time.
T	Local fluid temperature.
u	x component of velocity $\underline{v}$.

U	Weight function.
v	y - component of velocity $\underline{v}$.
$\underline{V}$	Local flow velocity vector.
w	z - component of velocity $\underline{v}$.
W_i	Weights for Gauss integration.
ω	Relaxation factor.
x,y,z	Global coordinate system.
α	Artificial viscosity coefficient.
γ	Ratio of specific heats.
ρ	Local fluid density.
$\tilde{\rho}$	Modified fluid density due to upwinding.
ξ,η,ζ	Local coordinate system for master element.
ϕ	Velocity potential function.
$\underline{\phi}$	Velocity potential vector.
$\underline{\tilde{\phi}}$	Relaxed velocity potential vector.
$\underline{\nabla}$	Gradient operator.
Ω	Flow domain.
Ω_e	Element domain.
π	Bateman-Dirichlet integral.

I. INTRODUCTION

1.1 STATEMENT OF THE PROGRAM

Analysis of flows around an aircraft at transonic speeds is of major importance in terms of its performance and maneuverability. At transonic speeds, local supersonic flow regions are usually terminated by weak, embedded shock waves. The subsonic-supersonic flows are extremely sensitive to the shape of the aircraft geometry. The mathematical difficulties related to the solution of this problem are associated primarily with the mixed, hyperbolic and elliptic type of equations for subsonic and supersonic flow fields and the presence of discontinuities called shock waves. A computational method, for analyzing transonic flows, should be capable of predicting the location and strength of these shock waves which are again very sensitive to the changes in the geometry of the boundaries.

A major concern, in the computation of transonic flows, is the design of a computational grid. A good computational grid is a basic requirement for an accurate flow analysis over a three-dimensional configuration. A surface-fitted grid permitting easy and exact introduction of boundary conditions on curved boundaries becomes necessary. While, in three-dimensional applications the number of points on the computational grid grows rapidly, description of complex geometries like wing-body combinations and description of boundary surface fitted grids become a difficult task and increase the labor involved.

During recent years, substantial progress has been made in the development of numerical methods for the solution of inviscid transonic flows. Some of the early studies to predict the inviscid flow fields around airfoils [1], wings [2] and simple wing-body combinations [3] started within the framework of small disturbance theory. The equations in this case were somewhat simpler and boundary conditions can easily be imposed on a mean surface. For complex geometries, this becomes too much of a simplification. Treatment of boundary conditions where boundary surface does not coincide with grid points is generally either complicated and time consuming or inaccurate. The need for the solution of full potential equation, rather than its small disturbance approximation, emphasized the requirements in applying the boundary conditions accurately in later finite difference calculations.

Jameson [4] solved transonic full-potential flow equation by rotating the difference scheme to conform with the local stream direction so that proper directional property is obtained. Following Jameson's work, many applications of both finite difference and finite volume methods were presented for the solution of full-potential equation. For analyzing flow around three-dimentional, complex configurations, these techniques required extension of mesh generation techniques. In order to represent curved boundaries, a regular finite difference grid, either an interpolation formula employed at grid points nearest the boundary surfaces to treat mesh-boundary intersections or a coordinate transformation was performed in order to reduce the boundaries to coordinate surfaces.

A major problem encountered in computational fluid dynamics is the generation and control of grids on which numerical solutions are to be obtained. The accuracy and convergence rate of a particular solution scheme generally depend on the degree of refinement and alignment of the grids with solution variables. Application of exact boundary conditions for the generated grid requires additionat attention as the flow field geometries become more complex.

In finite difference applications, a general method for constructing grids is the mapping of the physical flow domain onto a rectangular domain. The mapping transformation is represented as the solution to an elliptic boundary value problem for the rectangle [17]. In the case of complex three-dimensional geometries, the application of transformations become complicated and even limited. An alternate approach is to use a block-structured grids. In this approach, the computational domain is divided into multiple rectangular blocks that can be defined to produce surface fitted computational grids. In this case, the necessity of collapsing some block edges requires additional computations for representing complex geometries [18].

Many studies have also been conducted for the purpose of obtaining better orthogonality, introducing variable grid spacing and satisfying better alignment with the boundaries [19-21]. In addition, solution adaptive grids in which grid points are rearranged to improve accuracy have been introduced by several researchers [22,23].

As compared to the finite difference approach, finite element method handles the problem in the physical plane and uses only a local mapping during integrations in the computation. On the other hand, the description of the physical geometry and preparation of finite element grid still has to be defined in an efficient

manner. Finite element grid generation schemes which are commonly used for the solution of complex structural mechanics problems include many features to automate the data preparation. Yet, being general purpose programs, they require still considerable labor for solving three-dimensional aerodynamics problems.

In the present study, the main objective was to illustrate the application of finite element technique to generate a computational grid for analyzing the transonic flow around a complex three-dimensional configuration. In this report, first a solution technique for analyzing three-dimensional transonic flows using the finite element method is summarized; then, a finite element mesh generation scheme suitable for modeling wing-body configurations is presented. This procedure involves a two-level generation of finite element grids. First, the physical domain is represented with small number of isoparametric elements which are called blocks. These large blocks are fitted to the complex geometries. Then, each of the blocks are remeshed into finer elements automatically. This process requires a minimum amount of data to define the finite element grid. Finally, by using these finite element grids, transonic full-potential equation is solved for a sample problem. For this test case, the generation of an "optimum" grid is discussed. An iterative approach is proposed where an initial grid is modified for improving accuracy. It is shown that by using the developed grid generation scheme, one can do this efficiently and control the grid spacing at critical flow regions without disturbing the rest. Also, it is shown through the computational experiments performed in the present study, that the design of an optimum grid requires an understanding of the flow field which can be obtained through this iterative procedure.

1.2 TRANSONIC FLOW PROBLEM

1.2.1 Governing Equations

The governing equation for steady, inviscid and irrotational flow can be expressed in terms of velocity potential ϕ and density ρ considering conservation of mass and irrotationality of the fluid. Conservation of mass can be written as follows:

$$\nabla\cdot(\rho\underline{V}) = 0 \tag{1.2.1}$$

where the velocity vector is,

$$\underline{V} = u\underline{i} + v\underline{j} + w\underline{k} \quad . \tag{1.2.2}$$

By using the condition of irrotationality, the velocity vector can be written as,

$$\underline{V} = \nabla\phi \tag{1.2.3}$$

where the scalar function ϕ is the velocity potential. Equation (1.2.1), then becomes,

$$(\rho\phi_{,x})_{,x} + (\rho\phi_{,y})_{,y} + (\rho\phi_{,z})_{,z} = 0 \tag{1.2.4}$$

For three-dimensional flows, the solution of equation (1.2.4) should satisfy the following Neuman type flux boundary condition:

$$\rho\phi_{,n} = f \tag{1.2.5}$$

where f is the mass flux on the boundary surface whose outward normal is n. On the solid boundaries, f is assigned to be zero. On the other hand, since Neuman boundary conditions are involved, the potential will be dependent to an arbitrary constant, to remedy this, the value of ϕ at any point within the solution domain is to be prescribed [26].

1.2.2 Variational Formulation

The weak statement of the problem (1.2.4) is to determine the function ϕ such that differential equation and boundary conditions are satisfied in the sense of weighted averages:

$$\delta\pi = \int[(\rho\phi_{,x})_{,x} + (\rho\phi_{,y})_{,y} + (\rho\phi_{,z})_{,z}]\ U\ dV = 0 \tag{1.2.6}$$

where U is the weight function and the integration is over the whole domain. No derivatives of weight function appears in (1.2.6). On the other hand, the second derivatives of ϕ have to be calculated. An alternative symmetric weak formulation can be obtained by applying integration-by-parts and choosing U same as ϕ.

$$\int_S[(\rho\phi_{,x})\phi + (\rho\phi_{,y})\phi + (\rho\phi_{,z})\phi]\ dA$$

$$- \int_\Omega[(\rho\phi_{,x})\phi_{,x} + (\rho\phi_{,y})\phi_{,y} + (\rho\phi_{,z})\phi_{,z}]\ dV = 0 \tag{1.2.7}$$

The surface integral over S provides the natural boundary conditions for specified flux values. In the case of rigid walls, the normal mass flux $(\rho\phi_{,x_i})$ vanishes and natural boundary condition automatically provides zero normal flux conditions [14]. In the above formulation, if one can eliminate the variable density in terms of velocity potential, the problem can be written in terms of a single unknown, ϕ.

For isentropic flows, the relationship between density and local flow speed can be written in terms of velocity potential ϕ, as follows:

$$\rho = [\frac{\gamma-1}{2\gamma A} (q_{max}^2 - (\phi_{,x}^2 + \phi_{,y}^2 + \phi_{,z}^2))]^{1/(\gamma-1)} \quad (1.2.8)$$

where q_{max} is the maximum attainable velocity

1.2.3 Application of the Artificial Viscosity

The inspection of the weak solution in Equation (1.2.7) shows that the isentropic equations permit a possible discontinuity [27]. Since the potential equation is fully isentropic, it describes a reversible situation allowing expansion shocks as well as compression shocks. In the case of expansion shocks, entropy has to decrease, which is a violation of thermodynamics laws. In order to get correct weak solutions containing only physical shocks, an entropy condition has to be introduced in the modeling of the flow.

Murman and Cole [1] first demonstrated that shock waves can be obtained in the relaxation schemes if upwind differencing formulas are used in the supersonic regions. This is a correct differencing scheme in the sense that domain of dependence in the flow field is satisfied. In subsonic flow regions, the flow properties at a point are effected by the flow conditions all around the point. However, in supersonic zones, the flow properties depend only on the flow properties of the domain of dependence. The disturbances march away from an initial data plane and downstream influences cannot propogate upstream.

It can be shown that the error introduced by the application of backward differencing results is a second-order dissipation term which is analogous to the viscous terms in the Navier-Stokes equation. Thus, in order to solve transonic flow problems, when the flow is supersonic, the solution scheme must be modified to account for the hyperbolic nature of the governing equations. This can be done either by including explicity some artificial viscous terms at the supersonic locations or using difference schemes which introduce the second-order viscous term like truncation errors.

In this study, artificial viscosity like terms is introduced by using a modified density in the supersonic region. This density modification is given as [28]

$$\rho_e^n = \rho_e - \alpha_e \Delta s_e \rho_{e,s} \quad (1.2.9)$$

where s is the streamline direction, Δs is the element size in the direction of s and α is the coefficient of artificial viscosity. e stands for the element under consideration.

In the above formulation, since artificial viscosity is introduced by taking the element size into account, it satisfies uniform distribution of viscosity content. In the case of non-uniform grids, one has to consider the changing dimensions of the grid in the flow direction.

Gradient of density in equation (1.2.9) is obtained by backward differencing,

$$\rho_e^n = \rho_e - \alpha_e \Delta s_e \frac{(\rho_e - \rho_u)}{\Delta s_{eu}} \tag{1.2.10}$$

where,

$$\Delta s_{eu} = \tfrac{1}{2}(\Delta s_e + \Delta s_u) \tag{1.2.11}$$

and u stands for the nearby upstream element.

The choice of artificial viscosity is one of the important factors in determining the efficiency of the solution procedure. A previous study conducted on the choice of artificial viscosity distribution, which is based on a simple one-dimensional model [13], gives the following condition for uniform convergence:

$$\alpha_e > 1 - \frac{1}{M_e^2} \tag{1.2.12}$$

In the above expression, the effect of the position of the element in the supersonic pocket, the distribution of the error in the initial solution and size of the elements in the grid are not included in the evaluation of the artificial viscosity.

In this study, artificial viscosity coefficient is taken as;

$$\alpha_e = \mu[1 - \frac{1}{M_e^2}] \tag{1.2.13}$$

where, μ is the artificial viscosity content and it is constant during the iterations. Then, the equation (1.2.9) can be written as

$$\rho_e^n = \alpha_e^* \rho_u + (1 - \alpha_e^*)\rho_e \quad , \tag{1.2.14}$$

where

$$\alpha_e^* = \mu \frac{\Delta S_e}{\Delta S_{eu}}[1 - \frac{1}{M_e^2}] \tag{1.2.15}$$

II FINITE ELEMENT FORMULATION

2.1 FINITE ELEMENTS

Finite element formulation of the problem (2.1.7) is obtained by dividing the solution domain into smaller elements similar to the ones shown in figure (2.1). The potential function is then approximated within an element as a linear combination of its values at grid points based on the locally defined shape functions $N_i(x,y,z)$ assigned to each grid point:

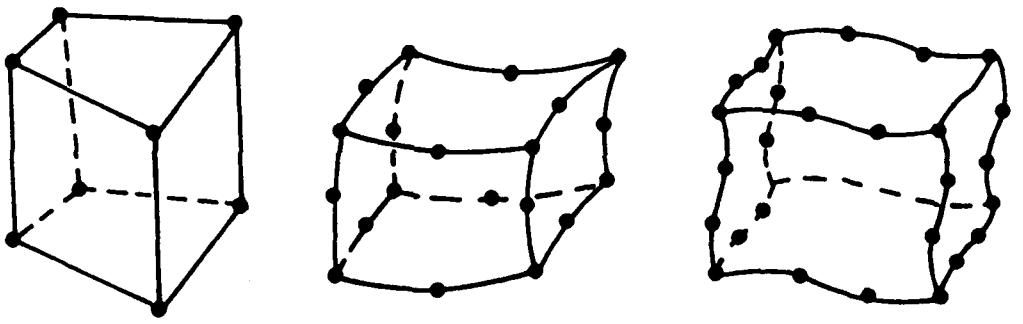


Fig. 2.1 Three-dimensional finite elements

$$\phi_e(x,y,z) = N_i(x,y,z)\phi_i \qquad (i=1,.....m) \qquad (2.1.1)$$

The number of approximation functions (shape functions) for each element is equal to the number of nodal points. In the above equation ϕ_i is the ϕ value at node i, N, is the shape function of node i and ϕ_e gives the distribution of ϕ inside the element e.

It follows from (2.1.1) that

$$N_I(x_i,y_j,z_k) = \begin{cases} 1 & \text{if } i=j=k=I \\ 0 & \text{otherwise} \end{cases} \tag{2.1.2}$$

where x_i is the x coordinate of the i^{th} node of the element and so on.

For an n noded element, the following column vectors can be defined:

$$\underline{\phi}_e^T = [\phi_1,\phi_2, \ldots\ldots \phi_n] \tag{2.1.3}$$

$$\underline{N}^T = [N_1,N_2, \ldots\ldots N_n] \tag{2.1.4}$$

Here, $\underline{\phi}_e$ is the element nodal vector of ϕ_i values corresponding to the nodal points and $\underline{N}$ is the element shape function vector of N_i, shape functions for every nodal point. The superscript T denotes the transpose of these vectors.

The equation (2.1.1) can be written in the following form using the above rotation as follows:

$$\phi_e\ (x,y,z) = \underline{N}^T\underline{\phi}_e \tag{2.1.5}$$

Substitution of equation (2.1.5) into (1.2.7) results in,

$$\underline{K}\ \underline{\phi} = \underline{F} \tag{2.1.6}$$

where

$$\underline{K} = \sum_e \int_{\Omega_e} \rho_e (\underline{N},_x \underline{N},_x^T + \underline{N},_y \underline{N},_y^T + \underline{N},_z \underline{N},_z^T)\ dV \tag{2.1.7}$$

$$\underline{F} = \sum_e \int_S f_e \underline{N} dS \qquad f_e = \text{element flux vector.} \tag{2.1.8}$$

$$\underline{\phi} = \sum_e \underline{\phi}_e \tag{2.1.9}$$

The coefficient matrix $\underline{K}$ is obtained as a result of contribution of element coefficient matrices for every node. Contributions of the surrounding elements are considered during the assembly of element matrices.

The specified flux boundary condition which is directly included in the variational formulation becomes the right-hand-side of the equation (2.1.6). Equation (2.1.8) is the surface integral expression written for every element over the element boundaries. Since the elements satisfy the continuity of mass, contributions of flux terms at the inter-element boundaries for the entire system cancel each other. At the interface, the same flux leaves the surface of one element and enters the other one, the resultant flux being equal to zero. In the

case of elements with a surface on the specified flux boundaries, equation (2.1.8) provides the application of natural boundary conditions.

2.1.1 Linear and Higher-Order Element Formulations

As it can be seen in the equation (2.1.7), the global x,y,z coordinates are used for the calculation of local approximations over each element. In actual computations, all these calculations are generally performed for a parent element in terms of local coordinates ξ, η, ζ and are transformed to the global coordinates for each element in the grid.

In order to evaluate the coefficient matrix (2.3.7) and load vector (2.3.8), two transformations are necessary. The first one is the expression of global derivatives in terms of the local derivatives. The second one is the expression of element volume and element surface over which the integrations have to be performed. Then, the integration is performed in terms of local coordinates with proper integration units.

This formulation allows an efficient formulation as well as the utilization of higher-order isoparametric elements. An isoparametric serendipity element can be defined as having linear, quadratic and cubic node configurations along its edges in an arbitrary manner as shown in Fig. 2.2.

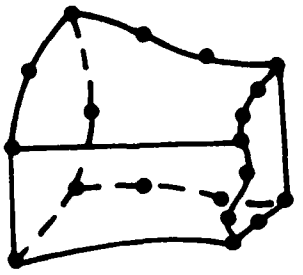


Fig. 2.2 Isoparametric serendipity element

Shape functions and their derivatives can be expressed in terms of multiplications of three, one-dimensional basic functions and their derivatives in the local coordinate system. In this coordinate system (ξ, η, ζ), each element in global coordinates shown in Fig. 2.2 with the local origin at the element centroid. The values of the local coordinates vary between the limits of + 1 and - 1.

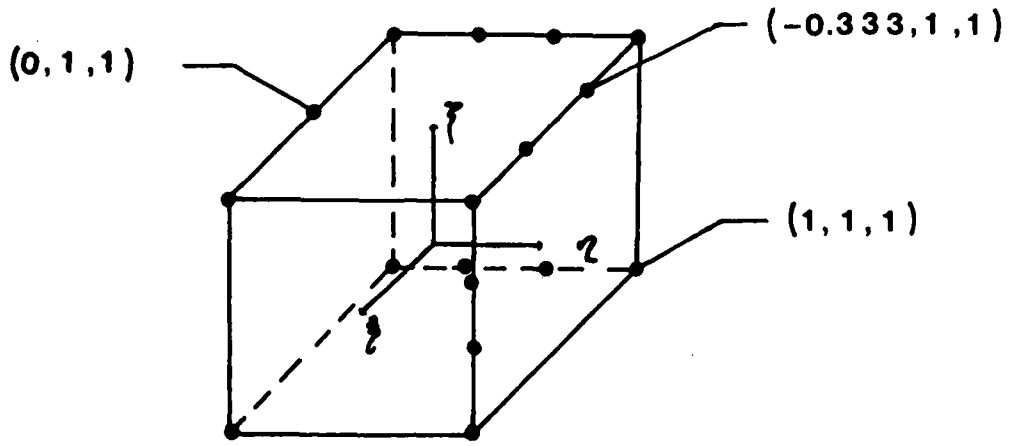


Fig. 2.3 Master serendipity element

In the parent element, for the linear case, nodes are at the corners, for parabolic elements, additional nodes are located at the mid-point of each edge, while for cubic elements, four nodes are located equally spaced on each edge.

In the case of a surface, any of the element in figure 2.2 is transformed into a perfect square by using two-dimensional shape function formulation. The same discussion in the above paragraph is valid for this case as well.

The above transformations can now be carried out by the use of shape functions. The coordinate transformation for an n noded element, from local to global, by using shape functions N_i, is given as follows:

$$x = N_i(\xi,\eta,\zeta)x_i \qquad (i=1 \ldots\ldots n)$$

$$y = N_i(\xi,\eta,\zeta)y_i \qquad (i=1 \ldots\ldots n) \qquad (2.1.10)$$

$$z = N_i(\xi,\eta,\zeta)z_i \qquad (i=1 \ldots\ldots n)$$

where x_i, y_i, z_i are the global coordinates of the corresponding i^{th} node. $(N_i, i = 1,n)$ are the shape functions calculated at the point defined by local coordinates ξ, η, ζ.

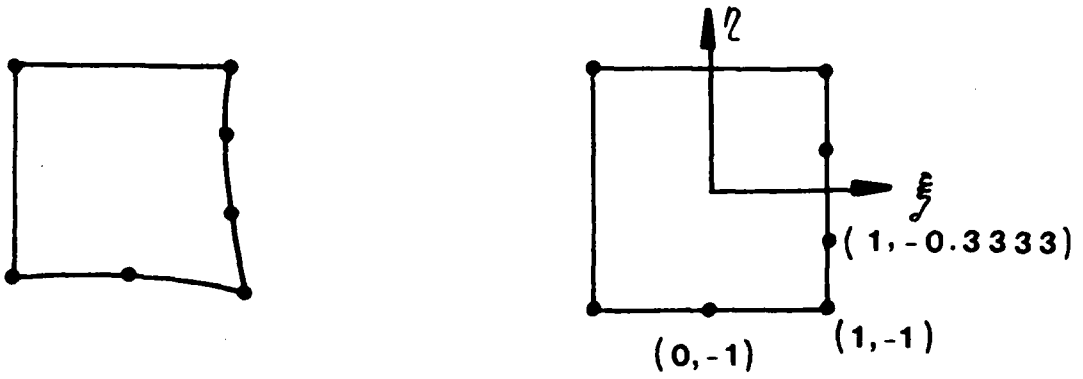


Fig. 2.4 Two-dimensional serendipity elements

Deriviatives of shape function in terms of global coordinates, are obtained by partial differenciation as follows:

$$N_{i,\xi} = N_{i,x}\, x_{,\xi} + N_{i,y}\, y_{,\xi} + N_{i,z}\, z_{,\xi} \qquad (2.1.11)$$

which can be repeated for η and ζ. In matrix form, this can be written as,

$$\begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \\ N_{i,\zeta} \end{bmatrix} = \begin{bmatrix} x_{,\xi} & y_{,\xi} & z_{,\xi} \\ x_{,\eta} & y_{,\eta} & z_{,\eta} \\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta} \end{bmatrix} \begin{bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{bmatrix} \qquad i=1....n$$

$$= |\underline{J}| \begin{bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{bmatrix} \qquad (2.1.12)$$

Derivatives of shape functions in terms of local coordinates, i.e. left-hand-side of the equation (2.1.12) can be evaluated at the given point. Furthermore, the explicit expression giving x, y, z in terms of ξ, η, ζ in equation (2.1.10) can be differentiated to obtain the matrix J, which stands for Jacobian matrix of transformation. In order to find the global derivatives of shape functions, the above system should be written in the following form:

$$\begin{bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{bmatrix} = |\underline{J}^{-1}| \begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \\ N_{i,\zeta} \end{bmatrix} \tag{2.1.13}$$

The necessary condition for the above system to be invertible is that the determinant of the Jacobian is not zero. Furthermore, it should be greater than zero, so that a unique and proper transformation is satisfied as defined in equation (2.1.13).

2.1.2 Integration Over the Volume of an Element

The volume integration seen in the equation (2.3.7) can now be evaluated over the parent element by introducing the determinant of the Jacobian and choosing the appropriate integration limits. A geometric interpretation of Jacobian indicates that the image of the volume $dV' = d\xi d\eta d\zeta$ has a volume dV in x,y,z where $dV = |J| d\xi d\eta d\zeta$ becomes

$$dV = dx(dyxdz) = \det \begin{bmatrix} x_{,\xi} & x_{,\eta} & x_{,\zeta} \\ y_{,\xi} & y_{,\eta} & y_{,\zeta} \\ z_{,\xi} & z_{,\eta} & z_{,\zeta} \end{bmatrix} d\xi d\eta d\zeta \tag{2.1.14}$$

The integration of the element in equation (2.3.7) can then be performed as follows:

$$K_e = \int_{\Omega_e} \rho_e [\underline{N}_{,x} \underline{N}_{,x}^T + \underline{N}_{,y} \underline{N}_{,y}^T + \underline{N}_{,z} \underline{N}_{,z}^T] \, dxdydz$$

$$= \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \rho_e [\underline{N}_{,x} \underline{N}_{,x}^T + \underline{N}_{,y} \underline{N}_{,y}^T + \underline{N}_{,z} \underline{N}_{,z}^T] |J_e| d\xi d\eta d\zeta \tag{2.1.15}$$

where the derivative shape functions are defined in terms of local element coordinates by equation (2.1.13).

The above integration is carried out numerically. A convenient way of doing this is to use Gaussian quadrature which are defined for regular geometric shapes such as the master elements used in finite element analysis. Gaussian quadrature is specified by a number of integration points each with an associated weight. The integration points and their weights are given in the appendix A for various order of integration. If the integrant is a polynomial, the integral can be calculated exactly by choosing the proper order.

Integration of the above system can then be obtained as:

$$K_e = [(\rho_e S_e |J|)_{P_i} W_i] \tag{2.1.16}$$

where:

$$\underline{S}_e = [\underline{N},_x \underline{N},_x^T + \underline{N},_y \underline{N},_y^T + \underline{N},_z \underline{N},_z^T] \tag{2.1.17}$$

P_i is the integration point with three local coordinates and (i=1,..m),m is the number of integration points.

Since the integrand is not a simple polynomial, selection of integration points is based on the following rules. Ideally as the element size goes to zero, the integrand reduces down to a constant value, det J, which at least, should be integrated exactly. The order of det J for linear, parabolic and cubic is 1, 5 and 8 respectively. Numerical experiments have shown that for the potential equation, 2 point integration for linear elements and 3 point integration for cubic elements give sufficient accuracy.

2.1.3 Integration over the Surface of an Element

The integration over the surface of the element in equation (2.1.8) is carried out in the same way as the volume integral. The surface defined in the global coordinate system can be transformed to the local system of a square by means of two-dimensional shape functions.

$$\begin{aligned} x &= N_i(\xi,\eta)x_i \\ y &= N_i(\xi,\eta)y_i \\ z &= N_i(\xi,\eta)z_i \end{aligned} \tag{2.1.18}$$

Since the integration is evaluated over the master surface, the determinant of the Jacobian matrix, which in this case gives the ratio of surfaces, has to be calculated. This is done in an average sense considering the following transformations:

$$\begin{bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{bmatrix} = \begin{bmatrix} x,_{\xi} & y,_{\xi} \\ x,_{\eta} & y,_{\eta} \end{bmatrix} \begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} = \begin{bmatrix} x,_{\xi} & z,_{\xi} \\ x,_{\eta} & y,_{\eta} \end{bmatrix} \begin{bmatrix} N_{i,x} \\ N_{i,z} \end{bmatrix} = \begin{bmatrix} y,_{\xi} & z,_{\xi} \\ y,_{\eta} & z,_{\eta} \end{bmatrix} \begin{bmatrix} N_{i,y} \\ N_{i,z} \end{bmatrix}$$

$$= |\underline{J}_{xy}| \begin{bmatrix} N_{i,x} \\ N_{i,y} \end{bmatrix} = |\underline{J}_{xz}| \begin{bmatrix} N_{i,x} \\ N_{i,z} \end{bmatrix} = |\underline{J}_{yz}| \begin{bmatrix} N_{i,y} \\ N_{i,z} \end{bmatrix} \qquad (2.1.19)$$

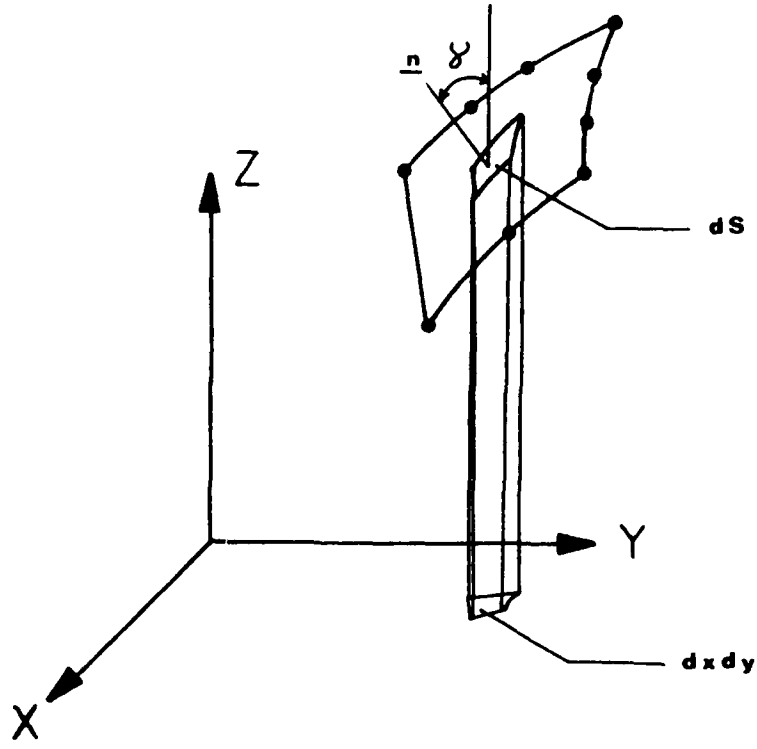


Fig. 2.5 Integration of the surface area of an element

As shown in the above figure; as the surface area goes to zero, it can be expressed in the following form:

$$dS = \sec\gamma dxdy \ ,$$

where γ is the angle between z axis and the normal direction of the surface,

$$\sec \gamma = \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1 \tag{2.1.20}$$

and from the definition

$$\begin{aligned} dxdy &= |J_{xy}| d\xi d\eta \\ dxdz &= |J_{xz}| d\xi d\eta \\ dydz &= |J_{yz}| d\xi d\eta \end{aligned} \tag{2.1.21}$$

The equation (2.1.21) leads to,

$$\frac{dz}{dy} = \frac{|J_{xz}|}{|J_{xy}|} \quad , \quad \frac{dz}{dx} = \frac{|J_{yz}|}{|J_{xy}|} \tag{2.1.22}$$

Combining the equations (2.1.20) and (2.1.22), one can write,

$$ds = [|J_{xz}|^2 + |J_{yz}|^2 + |J_{xy}|^2]^{\frac{1}{2}} \, d\xi d\eta$$

or

$$ds = |J_s| d\xi d\eta \tag{2.1.23}$$

where;

$$|J_s| = [|J_{xz}|^2 + |J_{yz}|^2 + |J_{xy}|^2]^{\frac{1}{2}} \tag{2.1.24}$$

The integration of the expression in equation (2.1.8) over the boundary surface becomes;

$$\int_S f_e \underline{N} ds = \int_{-1}^{1}\int_{-1}^{1} f_e \underline{N} |J_s| d\xi d\eta$$

Application of Gauss-integration rules result in;

$$\int_S f_e \underline{N} ds = [(f_e \underline{N} |J_s|)_{P_i} Wi] \tag{2.1.25}$$

where

$\underline{N}^T = \underline{N}^T(\xi,\eta) = \underline{N}_i$, $(i=1,n)$, n is the number of grid points on the surface.

In the present study, two point quadrature rule was used for linear and parabolic surfaces. In the case of cubic surfaces, three points were employed.

2.2 SOLUTION OF THE SYSTEM OF EQUATIONS

The system of algebraic equations, obtained in equation (2.1.6) in terms of nodal velocity potentials, is nonlinear and its solution requires an iterative process. One way to approach this problem is by treating it as a limit of a pseudo-unsteady problem, where the solution is obtained by time integration. It is stated in [29] that, although it does not correspond to the physical transient behaviour of the fluid, the steady-state solution can be reached by integrating such a pseudo-unsteady problem in time. In their studies, Ecer and et.al. have used such a pseudo-unsteady formulation and demonstrated its efficiency [13], [16], [28]. The same formulation will be used in this study.

It is assumed that the potential function and density are functions of time;

$$\phi = \phi(x,y,z,t) \qquad \text{and}$$

$$\rho = \rho(x,y,z,t) \tag{2.2.1}$$

and a steady-state is determined as the time step goes to infinity as follows;

$$\lim_{t\to\infty} \phi(x,y,z,t) = \phi_s(x,y,z)$$

$$\lim_{t\to\infty} \rho(x,y,z,t) = \rho(x,y,z) \tag{2.2.2}$$

With these assumptions, equation (2.3.6) can be written as:

$$\underline{K}(x,y,z,t)\phi(x,y,z,t) = \underline{F}(x,y,z) \tag{2.2.3}$$

In order to describe the problem as a pseudo-unsteady process, an artificial damping term is included in the equation such that it vanishes at the steady-state.

$$\frac{\Delta t}{\omega} \underline{K}^o \underline{\phi}_{,t} + \underline{K}\,\phi = \underline{F} \tag{2.2.4}$$

where Δt is the time step, ω is the relaxation factor and $\underline{K}^o$ is a constant coefficient matrix. It should be noted that as,

$$\lim_{t\to\infty} \phi_{,t} = 0$$

steady solutions of equations (2.1.6) and (2.2.4) are the same.

At the time step n, equation (2.2.4) can be written as,

$$\frac{\Delta t}{\omega} \underline{K}^{o} \underline{\phi}_{,t}^{n} + \underline{K}^{n} \underline{\phi}^{n} = \underline{F}^{n} \tag{2.2.5}$$

The time derivative of the potential function in the above equation is evaluated by using forward differencing in time.

$$\phi_{,t}^{n} = \frac{\underline{\Phi}^{n+1} - \underline{\phi}^{n}}{\Delta t} \tag{2.2.6}$$

On the other hand, in order to be able to control the convergence and stability of the problem, a relaxation factor is introduced. The solution obtained at n^{th} iteration step, $\underline{\Phi}^{n+1}$, is relaxed for the next iteration step by use of a relaxation parameter ω in the following way;

$$\underline{\phi}^{n+1} = \omega(\underline{\Phi}^{n+1}) + (1-\omega)\underline{\phi}^{n} \tag{2.2.7}$$

where $\underline{\Phi}^{n+1}$ is the solution of finite element equations at $(n+1)^{th}$ step and $\underline{\phi}^{n+1}$ is the relaxed solution which will be used for the next iteration.

Substitution of (2.3.7) into the equation (2.2.6) results in;

$$\underline{\phi}_{,t}^{n} = (\underline{\Phi}^{n+1} - \underline{\phi}^{n}) \frac{\omega}{\Delta t} \tag{2.2.8}$$

and the equation (2.2.5) reduces to,

$$K^{o} \underline{\Phi}^{n+1} = \underline{F}^{n} + (\underline{K}^{o} - \underline{K}^{n}) \underline{\phi}^{n} \tag{2.2.9}$$

The above pseudo-unsteady system of equations can be written as a system of algebraic equations in the following form:

$$\underline{K}^{o} \underline{\Phi}^{n+1} = \underline{F}^{*} \tag{2.2.10}$$

where

$$\underline{K}^{o} = \sum_{e} \underline{K}_{e}^{o}(x,y,z) \tag{2.2.11}$$

$$\underline{\Phi}^{n+1} = \sum_{e} (\phi_{e}(x,y,z))^{n+1} \tag{2.2.12}$$

$$\underline{F}^{*} = \sum_{e} \{\underline{F}_{e}(x,y,z) + [[\underline{K}_{e}^{o}(x,y,z) - \underline{K}_{e}^{n}(x,y,z,t)]\phi_{e}^{n}(x,y,z,t)]\} \tag{2.2.13}$$

The advantage of the numerical integration scheme based on the utilization of a constant-coefficient matrix $\underline{K}^o$ is obvious. A full decomposition of the coefficient matrix $\underline{K}^o$ is needed only for the first time step, while the subsequent iterations can be performed with forward and backward substitutions. If a variable coefficient matrix was chosen as $\underline{K}^n$, it would need to be reassembled and decomposed at every iteration step, which makes the scheme computationaly inefficient. The importance of the selection of $\underline{K}^o$ as a function of ρ^∞;

$$\underline{K}^o = \sum_e \iint_{\Omega_e} \rho^\infty S_e(x,y,z)d\Omega \tag{2.2.13}$$

gives comparable rates of convergence with a variable coefficient scheme.

In the previous applications of the finite element method, it was also observed that the accuracy of the solution and the rate of convergence was strongly dependent on the amount of the artificial viscosity content and the relaxation parameter. Another study [14] showed the effects of the relaxation parameter on the rate of convergence. Since an explicit time integration scheme in the supersonic regions must satisfy Courant condition, relaxation parameter ω must be chosen inversity proportional to the amount of added artificial viscosity:

$$\omega < \frac{1}{\alpha_e M_e^2} \tag{2.2.14}$$

Although no under-relaxation is required in the subsonic regions, it was observed that use of different relaxation factors can cause numerical problems around the boundaries of the sonic pocket. An equal under-relaxation was used in the present study in both regions at the expense of decreasing the overall convergence rate of the numerical scheme.

The numerical treatment of the problem follows, in general, the same approach tried by Ecer and et.al. in their studies [13,14],[16]. Since the required artificial viscosity depends on the distribution of the error in the initial solution, a high value of artificial viscosity is introduced at the befinning in order to have a uniform convergence. The initial guess is taken as the incompressible flow solution. After having a converged result for this case, artificial viscosity content is reduced. Taking the previous result being the initial guess, iterations are then continued until the converged results with a minimum amount of artificial viscosity is obtained. A detailed investigation of artificial viscosity effects was conducted as a part of the present project and presented in reference [15].

III. GENERATION OF THREE-DIMENSIONAL FINITE ELEMENT GRIDS

3.1 INTRODUCTION

The finite element grid generation scheme, presented in this study, utilizes a multiple-block structure. A smooth and surface-fitted grid is produced for each of the blocks. The blocks are then assembled with these subgrid systems to form the computational grid for the complete configuration. The main objectives in the development of such a grid generation scheme were the following:

1. The user has complete control in addressing a critical flow region. For example, leading edge of a wing is located around a particular block. The user can design a mesh in this region by only considering the distribution of the elements in this single block.
2. The block structure is defined in a simple manner, yet can handle any complex geometry.
3. Re-meshing of each block can be done separately. The changes in each block is transferred to its neighboring blocks automatically.

The details of the developed proceedings are summarized in the following sections.

Each block system is defined as a large finite element. These elements can be linear, quadratic or cubic elements depending on the complexity of the boundary surfaces to be represented. They only have to approximately represent the boundary surface at this point. These are the same type of elements used in finite element calculations as shown in Fig. 2.2, and in reference [33]. The boundary surfaces of these blocks are either part of the domain boundaries or inter-block boundaries. Generation of elements inside a block is based on the pre-specified element densities along three principal directions and gradients along 12 edges.

Calculation of corresponding nodal point coordinates are carried out in a local block coordinate system, which is a perfect cube with corner nodes at 1, -1 locations in the (ξ,η,ζ) local coordinate system. By using shape functions, the computed values are then transformed to the global coordinate system. Nodes generated on the boundary surfaces are then relocated at the user defined boundary surfaces to match exactly to the boundary surface.

Generation of the element connectivity information, on the other hand, also, becomes an easy task by using the edge description of the elements efficiently. Connectivity data is calculated each time and only the connectivity of higher-order elements if present is stored permanently. Although the global connectivity of elements are obtained under the assumption that all blocks are connected to the others, special situations like the voids in the solution domains of radial type grids can be handled by using slit and coupled surface capabilities.

3.2 GRID GENERATION SCHEME

3.2.1 Definition of Blocks

The first step is the definition of natural coordinate system identified as I, J, K to describe the block structure. Block and element topologies, including their connectivity, numbering and gradient definitions are based on this natural coordinate system.

Figures 3.1 and 3.2, give an example of block definition in a physical problem and its representation in the natural coordinate system. Each block is identified by eight block corner nodes, plus one or two additional nodes for quadratic or cubic approximations, respectively, along each curved edge of the block. Selection of these nodes depends on the smoothness of the surface which will be represented by four edges. For highly curved or irregular surfaces, either a cubic edge representation should be used or the block numbers on the surface should be increased to provide relatively smooth block surfaces. No intermediate edge nodes are required for flat planes and most of the time for the inter-block surfaces.

One important point in defining the higher-order block structure is the determination of the proper locations of the intermediate edge nodes. In the isoparametric transformation, quadratic or cubic serendipity shape functions are used to approximate curved boundaries. In accordance with the definition of shape functions, the additional nodes for each edge should be located at one-half or one-third length of the actual curved edge for parabolic and cubic edges respectively. If the proper points are not selected, the shape of the curve may not be close to the actual shape of the boundary while the points chosen to define the curve lie on the boundary. Even though the boundary nodes can be relocated to the actual surface, too much deviation between the two curves might result in distorted elements. Also, the distribution of the nodes generated inside the block is effected in such cases.

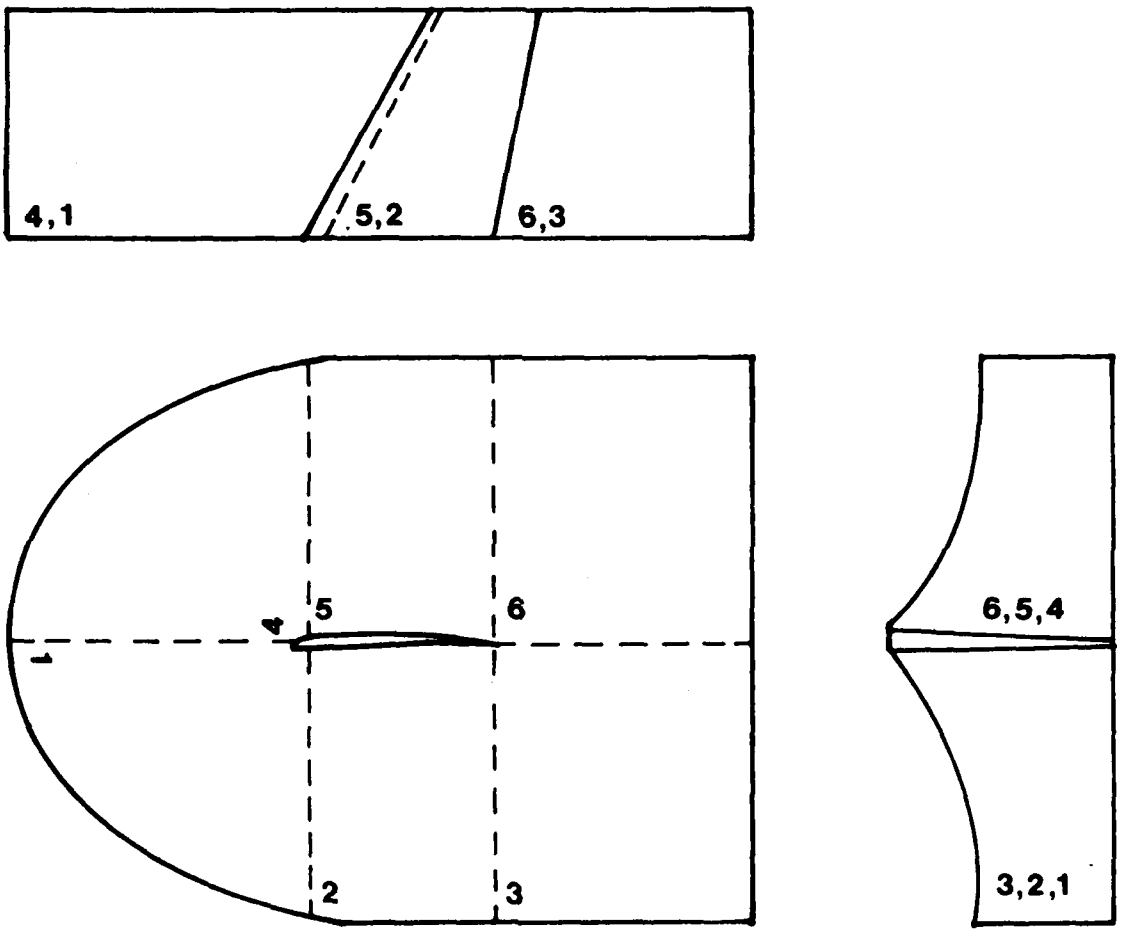


Fig. 3.1 A block structure definition in the physical domain

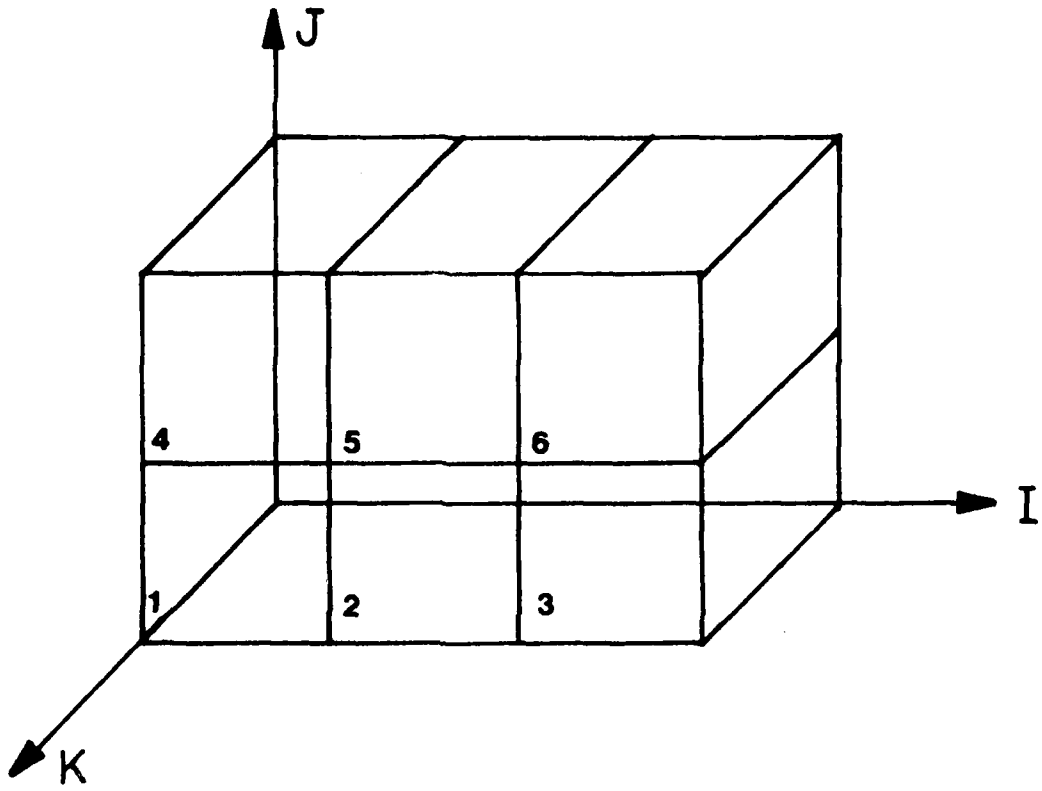


Fig. 3.2 The representation of the block structure in the natural coordinate system

In the mesh generation scheme, the three-dimensional domain to be modeled is divided into blocks by specifying the boundary surfaces, element densities and gradients together with any existing coupled surfaces and voids. The blocks are then numbered in I, J, K natural coordinate directions, starting with one and giving the increments along the I direction first and J and K sequentially. Such a numbering scheme provides the use of following formula which gives the global block number in terms of block natural coordinates IB, JB and KB and vice versa.

$$NBLOCK = IB + (JB-1)TNBI + (KB-1)TNBIJ \qquad (3.2.1)$$

where NBLOCK is the block number to be evaluated. TNBI and TNBJ are the total number of blocks in I and J directions respectively and TNBIJ is the total number of blocks in IJ plane (TNBI*TNBJ).

A block may be defined by a total number of nodes varying between 8 and 32 where 32 represents a case with all cubic edges. Although the global block node numbers can be assigned in any convenient way starting with 1, their connectivity must be associated with the local block node numbering used in the definition of shape functions. Local element node numbering which corresponds to the definition of the shape functions is shown in Fig. 3.3 with respect to local element coordinate system I′, J′, K′, obtained by translating the natural coordinate system to the local node 1 of the element. The same figure also shows local surface numbering which is used to specify slit, coupling and boundary surfaces which will be mentioned later in this chapter.

3.2.2 Generation of Elements and Nodal Coordinates

The distribution of elements inside each block is defined by two parameters. One of them is the element density which determines the number of elements in three principal directions and the other one is the element gradients which are specified for each edge of the block. These gradients are defined as the ratio of the first element length to that of the last element in the direction of natural coordinates.

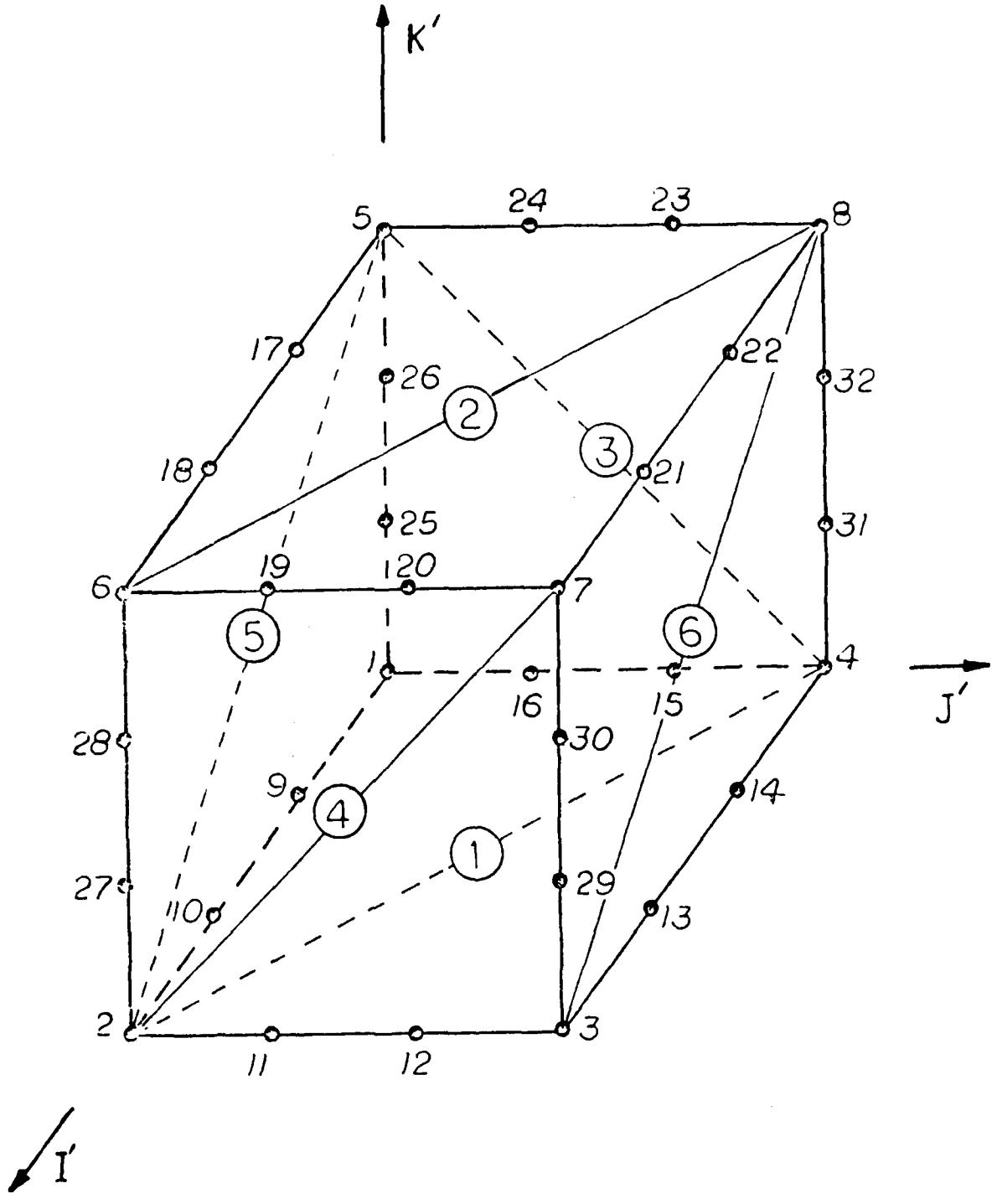


Fig. 3.3 Local node and surface numbers for a block

Each block is represented by a parent element which is a cube having the edge lengths of two units. A local coordinate system (ξ,η,ζ) is considered at the center point of the parent element as shown in Fig. 3.4.

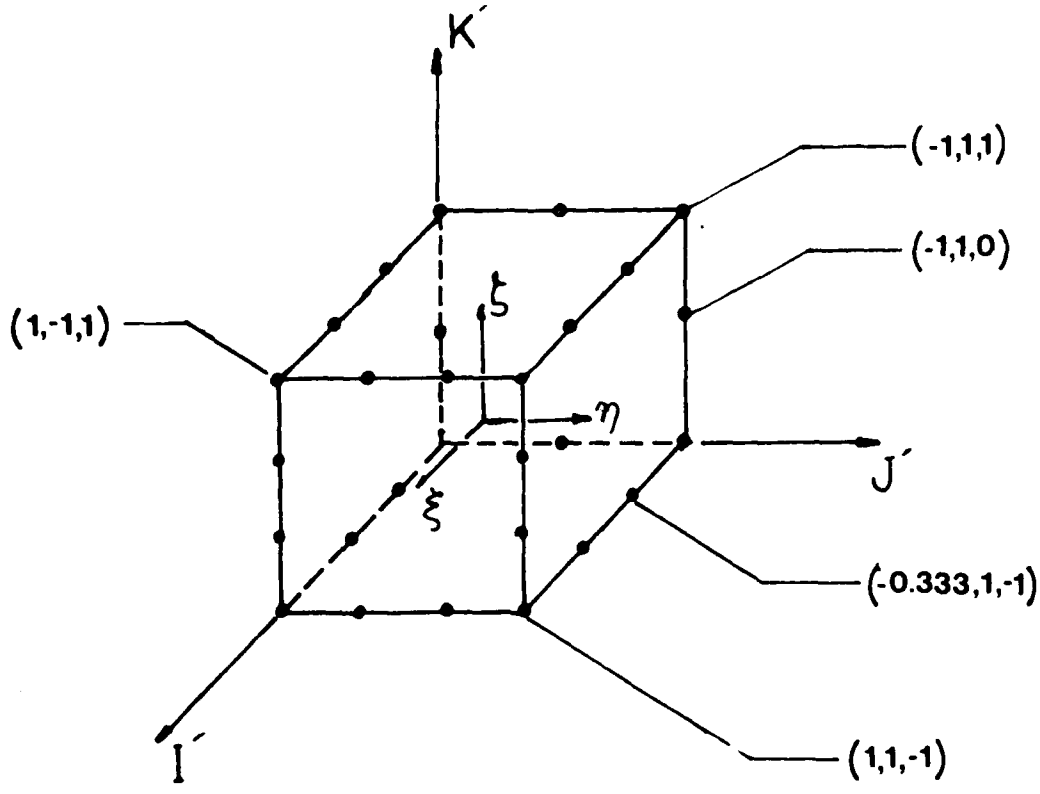


Fig. 3.4 Parent element

The main advantage of a mesh generation of mapping is utilized at this point. First, parent element coordinates for the corner nodes of each element in the block are generated by using element density D and gradient G. Division of elements started with the division of edges by assuming a geometric progression along the edge as shown in Fig. 3.5.

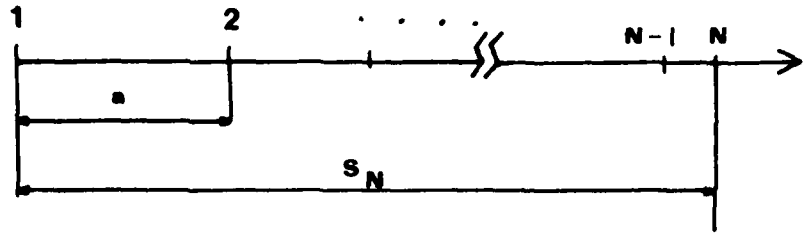


Fig. 3.5 Definition of Geometric progression

Geometric progression is defined by the following series;

$$S = a + ar + ar^2 = \ldots ar^n \tag{3.2.2}$$

where a is a constant which specifies the length of the first interval, r is another constant which gives the rate of progression, and n can be interpreted as number of nodal points along an edge which is equal to element density plus one.

Definition of gradient gives the following relationship,

$$G = \frac{S_n - S_{n-1}}{a} \quad . \tag{3.2.3}$$

For the geometric series, the partial sum of the terms is given as;

$$S_n = \frac{a(1-r^n)}{1-r} \quad .$$

Then, one can calculate the gradient as

$$G = \frac{a(1-r^n)-a(1-r^{n-1})}{(1-r)a} \quad .$$

After some manipulations, one can write

$$\begin{aligned} r &= \exp[\ln G/n-1] \\ &= \exp[\ln G/b] \quad . \end{aligned} \tag{3.2.4}$$

If the length of the edge is taken as 1, knowing r, a can be evaluated as follows:

$$S_n = \frac{a(1-r^n)}{1-r}$$

$$a = \frac{1-r}{(1-r^n)} = \frac{1-r}{1-r^{(E+1)}} \quad . \qquad (3.2.5)$$

In the parent element, an edge in natural coordinate directions is represented by the one-dimensional parent line having a length of two from -1 to 1. If the divisions on a unit length is transformed into the edge; one can, then write,

$$\xi = -1 + 2S_n \qquad n=1 \ldots\ldots \quad . \qquad (3.2.6)$$

Element and nodal point coordinate generation inside each of the blocks is obtained by using the divisions along the edges and interpolating linearly among them.

In the case of higher-order elements, the generation of intermediate edge node coordinates are based on the element corner node coordinates. This can be accomplished simply by evaluating the one-third or one-half locations, for cubic and parabolic edges respectively, along the edge. This simple interpolation which assumes a straight line edge between the corner nodes is applied in the global coordinates after the isoparametric transformation of corner node coordinates is done.

(ξ,η,ζ) coordinates of corner nodes generated in the parent block are then transformed by using the serendipity shape functions with the nodal coordinates defining the block. This is the same transformation used in finite element calculations in Eq. (2.1.10).

3.2.3 Generation of Node Numbers and Element Connectivity

In a finite element analysis, the preparation of element data, particularly element connectivity and its storage in the computer is an important task, especially, for analysing complete three-dimensional problems. In the present study node numbers are generated automatically and the connectivities of the nodes for each element determined, without storing them permanently.

In order to be able to determine the element number in terms of natural element coordinates and the node numbers in terms of element number, a sequentiaal numbering of elements and nodes in I, J, and K natural coordinates are employed. The element numbering is the same as that of blocks. The element number one is

assigned to the one located at the origin of I, J, and K natural coordinate system and numbering continues with the increment of one along the elements, in I direction. After the last element in I direction is numbered, element numbering in J direction and finally K direction is incrementally performed.

This numbering system implies that the elements in the natural IJ plane represents the wavefront.

The same formula as (3.2.1) applies for the determination of the element number in terms of natural coordinates IEL, JEL, KEL

$$NEL = IEL + (JEL - 1)TNELI + (KEL - 1)TNELIJ \qquad (3.2.7)$$

where NEL is the element number, TNELI, total number of elements in D direction, TNELIJ is the total number of elements in IJ plane.

Numbering of the element corner nodes is also based on the same approach. Node numbers are generated for the whole grid in I, J, K natural coordinate directions. The first node is the one at the origin and then the numbers with increment one are assigned to the nodes along the positive direction of natural coordinate axis I, sequentially. After the last node, assignment of node numbers moves to the second, third etc. rows along the positive J direction until the nodes on the first IJ plane are numbered. Afterwards the same process is applied to the second and third IJ planes along the positive K direction. An example of this scheme is given in Fig. 3.6. In this case total 4 blocks 2 in K direction are connected simply and their element density in J direction is 2.

The above numbering scheme leads to the use of following formulas for the evaluation of the corner nodes of an element located at IEL, JEL and KEL natural coordinates;

$$N1 = IEL \quad (JEL-1)TNODI + (KEL-1)TNODIJ \qquad (3.2.8)$$

$$N2 = N1+1$$

$$N3 = N2+TNODI$$

$$N4 = N1+TNODI$$

$$N5 = N1+TNODIJ$$

$$N6 = N2+TNODIJ$$

$$N7 = N3+TNODIJ$$

$$N8 = N4+TNODIJ$$

where N1 through N8 are the global element corner node numbers associated with the local node numbers 1 to 8 of the element shown in Fig. 3.3

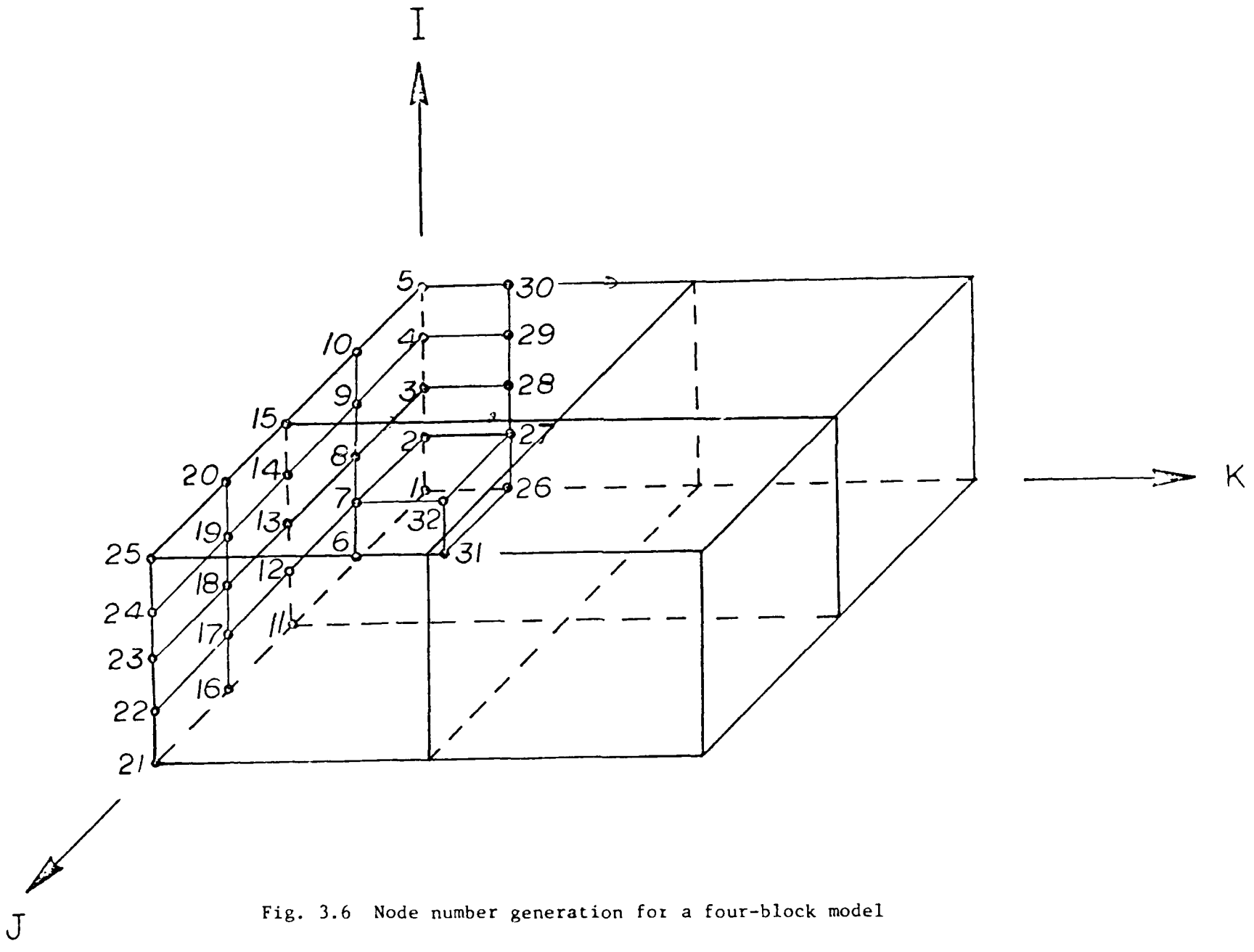


Fig. 3.6 Node number generation for a four-block model

TNODI, TNODJ are the total number of nodes in I and J natural coordinate directions of the nodel respectively. TNODIJ is multiplication of these two numbers which gives the number of nodes in the IJ natural plane.

The proper use of equations (3.2.7) and (3.2.8) provides the element corner connectivity in terms of global element number, NEL. It should be noted that equation (3.2.7) can be solved to determine IEL, JEL, and KEL values automatically if NEL is given.

For the case of higher-order elements, the element connectivity is again generated automatically by the program but this time stored in an array for each of the higher-order elements. Global element numbers are specified for each of the higher-order elements as input data. Then, the program starts processing this data by generating the intermediate edge node numbers for all specified quadratic and cubic elements. The maximum node number computed by assuming all elements to be linear is incremented and the element connectivity array is generated in accordance with the same configuration of nodes as in the definition of shape functions for the parent element Fig. 3.3.

Such an element specified as higher-order is connected to, at most, 6 other elements by face and 12 others by edge as shown in Fig. 3.7 and Fig. 3.8 respectively.

Since an edge which is common for 4 elements is to have the same connectivity for all these elements, when a higher-order element is specified, the connectivity arrays of the connected elements must be updated accordingly. This is also done automatically. First edge and then face connected element numbers are evaluated by exploiting the global element numbering scheme. Then, for these elements, their connectivity arrays are updated, considering the relative positions of the edges in each element. If an element which is already updated because of neighboring higher-order elements, it is processed later as a higher-order element. The new numbers are generated only for the edges which are not already updated to be of higher-order.

It should be noted that intermediate node numbers are not transparent to the user. They are controlled by the sequence of higher-order element specification list given in input data. Element number is sufficient in order to obtain the full element connectivity. Therefore, user has to be concerned with the element numbers and the order of the elements only.

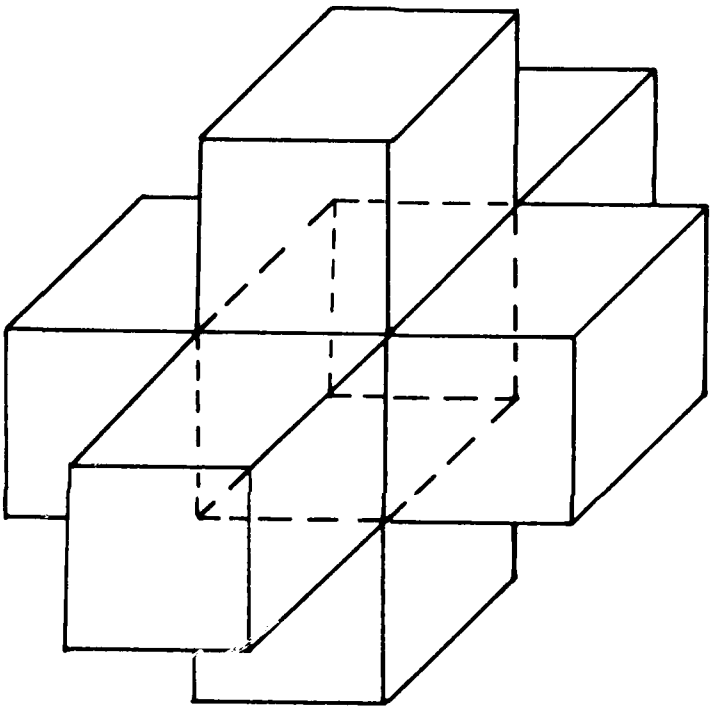


Fig. 3.7 Face connected elements

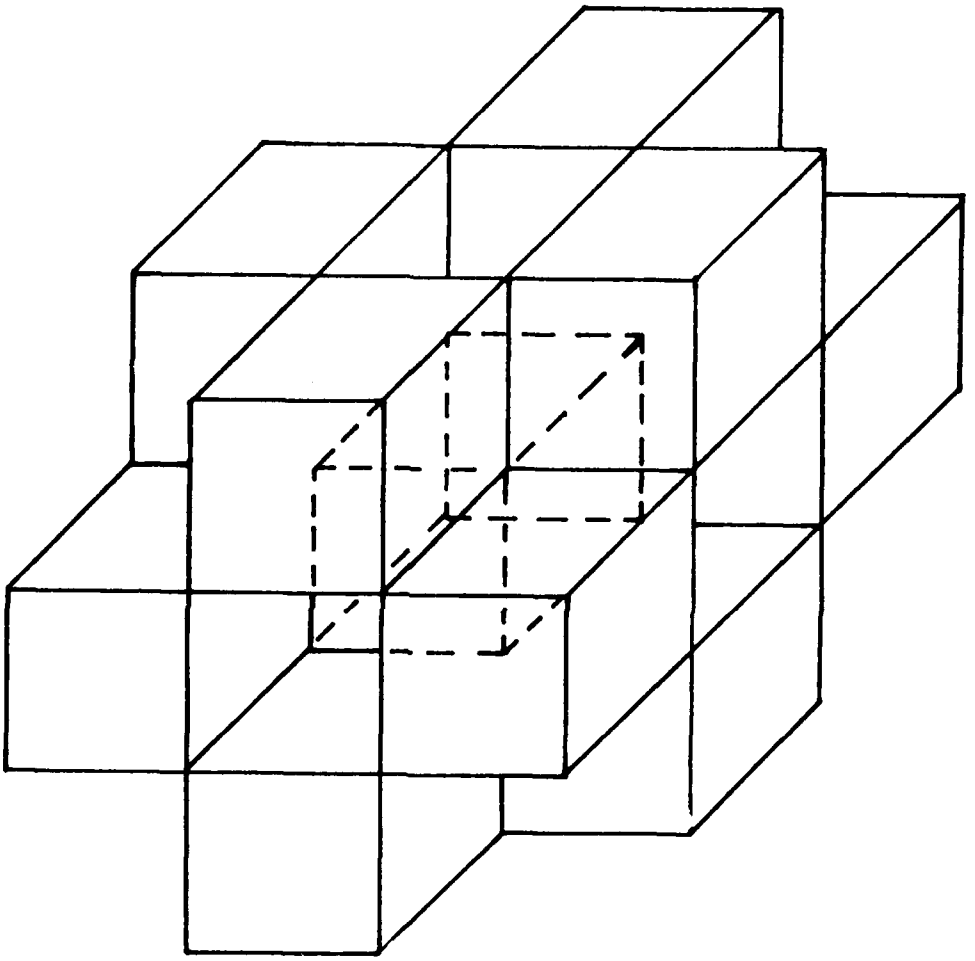


Fig. 3.8 Edge connected elements

3.3 SLIT AND COUPLED SURFACES

As it is stated earlier in this chapter in the generation of global element connectivities, it is assumed that the blocks are connected to each other and node numbers are considered to be the same for the nodes on the common faces of the blocks. The Fig. 3.9 illustrates the slit and couples surface conditons for modeling an aircraft wing in two-dimensions. The same concept can be extended to three-dimensions easily.

As it is seen in Fig. 3.9, the elements on the block surfaces which lie on the wing surface should not have the same surface connectivity as the elements on the opposite side of the wing. In this case, a wing can be placed between the blocks by simply disconnecting the neighboring blocks and assigning independent node numbers on the adjacent surfaces. In this manner, a slit can be placed between any neighboring blocks. These slits can also be used to impose Kutta-condition by allowing a potential jump between the neighboring surfaces. In this case the nodes on both sides of the slit occupy the same point in space. To distinguish two surfaces on each side of a slit, they are defined as master and slave surfaces. A master slit surface is defined by means of a block and natural surface number. Different corner node numbers are then assigned to the nodes on the slave surface. These additional nodes on the slave surfaces are numbered by incrementing the current highest node number. In the program, the four edges of the slit surfaces of corresponding blocks are also checked to determine if they have common edges. In this case, no additional generation is done. As it is seen in Fig. 3.9a, the edges in I direction, represented by nodal points 1, 13, are in common, thus, the elements sharing this edge has the same connectivity on this edge. It is also obvious that element numbers are not altered by existence of the slit surfaces.

If an element on one of the slit surfaces is defined to be of higher-order, after generation of regular new element nodes, existence of the slit is taken into consideration during the updating process. The edges on the slip surface, which are not common with the other surface, does not update the corresponding edges of face or edge connected elements.

The coupling of the surfaces is almost the reverse of the slit process. In this case, two different block surfaces which are not connected to each other in the nature of block definition, are connected and the same connectivity of the edges assigned. The Fig. 3.9 also shows the application of coupling. The generation of C type grids, as shown in this figure, or O type grids require a radial type of nodal layout. As in the case of slit surface definition, coupled

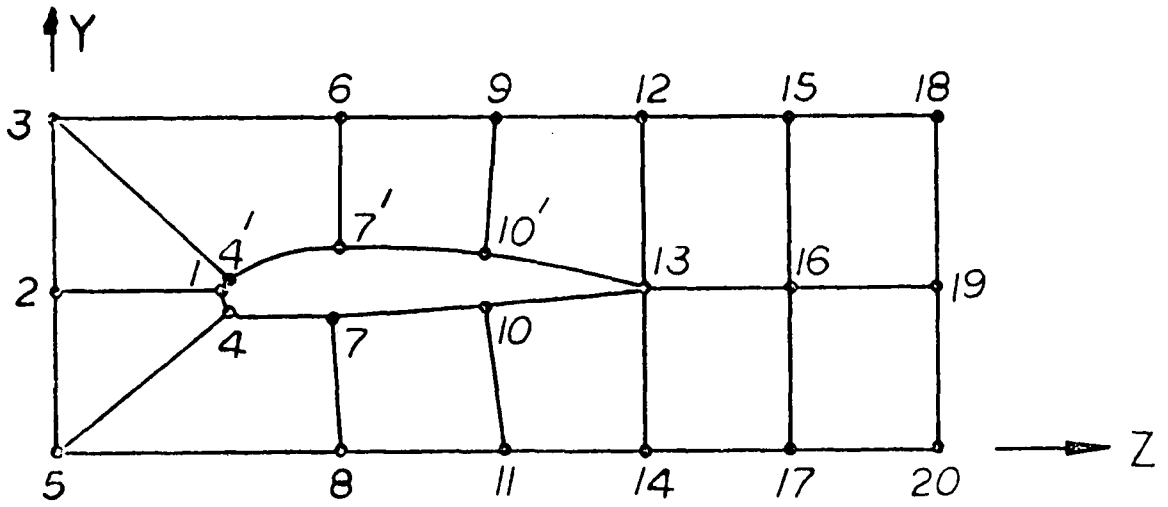


Fig. 3.9a Slit and coupled surfaces in the physical domain

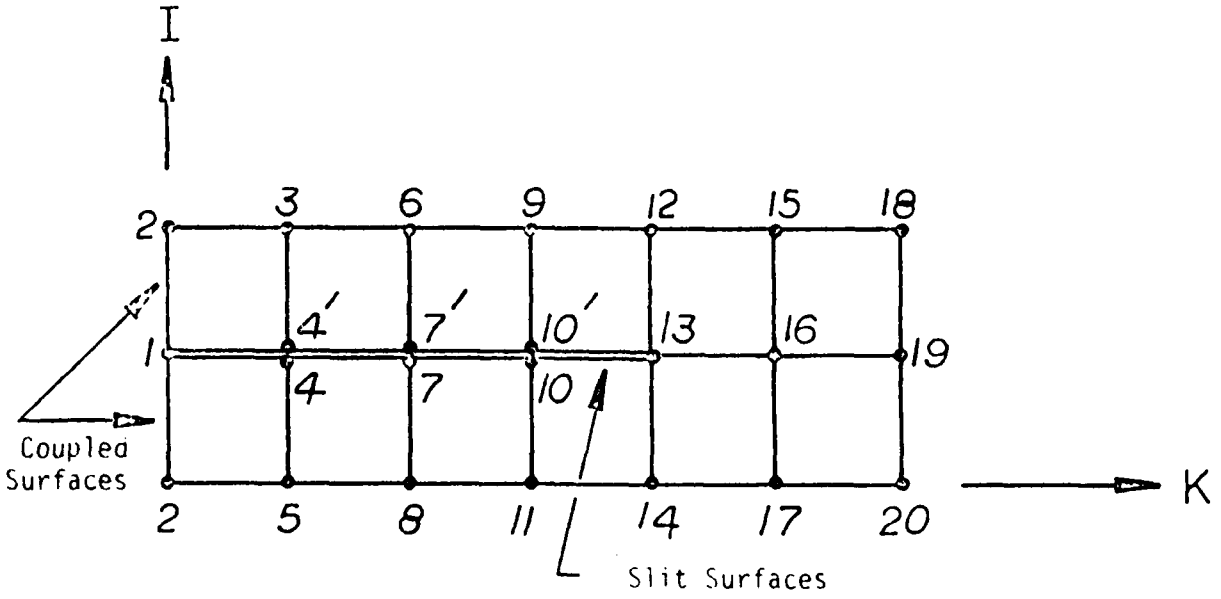


Fig. 3.9b Slit and coupled surfaces in the natural coordinate system

surfaces are defined by the global block number and the natural surface number and also by specifying one of the blocks as a master and the other as a slave.

During the evaluation of the linear connectivity of elements by using equations (3.2.8), every element is first checked, to determine whether it is in a block which has a slave coupled surface. If this is the case, then, it is further checked to find if it has a surface lying on the slave surface. When these conditions are met, instead of employing the equations (3.2.8) throughout the element, four nodes on the slave surface are assigned with the corresponding master element corner nodes.

When it comes to higher-order elements, coupling is the last process. After the higher-order connectivity and coordinate generation and updating by assuming regular connectivity of blocks, the node numbers are modified for coupling. The intermediate element edge nodes on the slave surface are assigned to the corresponding master edge nodes regardless of having any node number generated or updated before. Assignment sweeps all the elements on the slave surface and checks if the corresponding master element is of higher-order.

Although the concept behind the handling of slit and coupled surfaces is quite simple, it takes considerable computational effort.

3.4 BOUNDARY SURFACES

3.4.1 Introduction

The basic idea of the three-dimensional grid generation, as stated earlier, is the use of interconnected blocks which cover the overall physical domain. In this case; domain boundary surfaces specify the boundary surfaces of these blocks facing the domain boundaries. They can be defined as input by the coordinates of the corner points of the area to be described and one or two points along each curved edges of a surface. As it is also stated before, one of the difficulties encountered in this type of input is the selection of proper points to define these curved edges. If the points are not properly chosen, the shape of the surface produced may not be a good approximation to the actual shape of the boundary. The points chosen to define the surface will lie on the boundary surface. However, the element nodes on the same boundary surface of the block which are generated by using shape functions corresponding to the points defining the boundary, may provide a poor representation of the actual boundary.

The smoothness of the boundary surfaces may be lost without the knowledge of the user. In general, the accuracy in the solution of physical problems, especially problems whose solutions are determined by elliptic partial differ-

ential equations to a great extend, depend on the application of boundary concitions, which requires a description of actual physical boundaries. In numerical analysis, this is accomplished by having all the grid points representing physical boundaries to be on the actual physical surface.

In this study, the approach to this particular problem in the relocation of the boundary surface grid points generated by isoparametric transformation, to the actual physical surface which is to be defined separately. As it is explained in Fig. 3.10, the relocation process is basically determination of an intersection point on the boundary surface defined with a line which is normal to the isoparametric surface at the nodal point to be relocated.

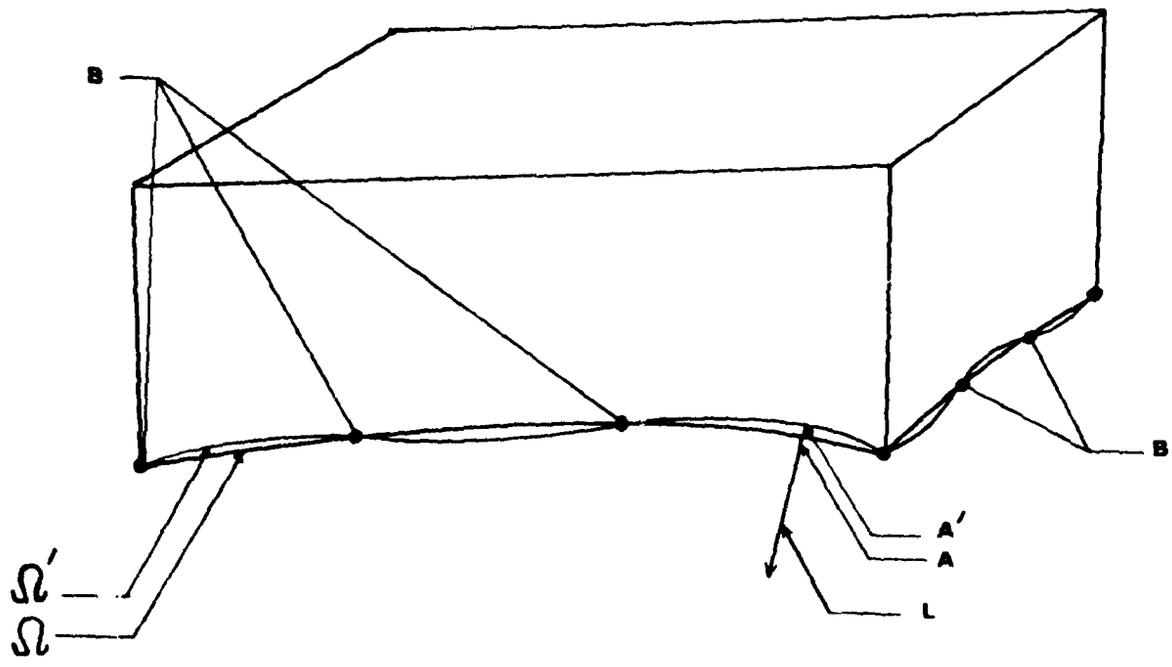


Fig. 3.10 Isoparametric surface definition for a block

In this figure, Ω' is the surface generated by isoparametric transformation based on the block input nodes denoted by B. The element corner nodes generated inside the block, denoted by A', could be considerable off from the actual surface, Ω. If a line, L which passes through the point A' and normal to the surface Ω' at this point is determined and its intersection with the actual surface, A is evaluated, the point A can be taken as the best approximation satisfying the actual boundary condition and the element gradient specified in the block.

The same relocation process is carried out for intermediate nodes of higher-order elements. In this case the element surface itself represents the isoparametric surface and intermediate nodes generated by linear interpolation are relocated on the physical surface as shown in Fig. 3.11.

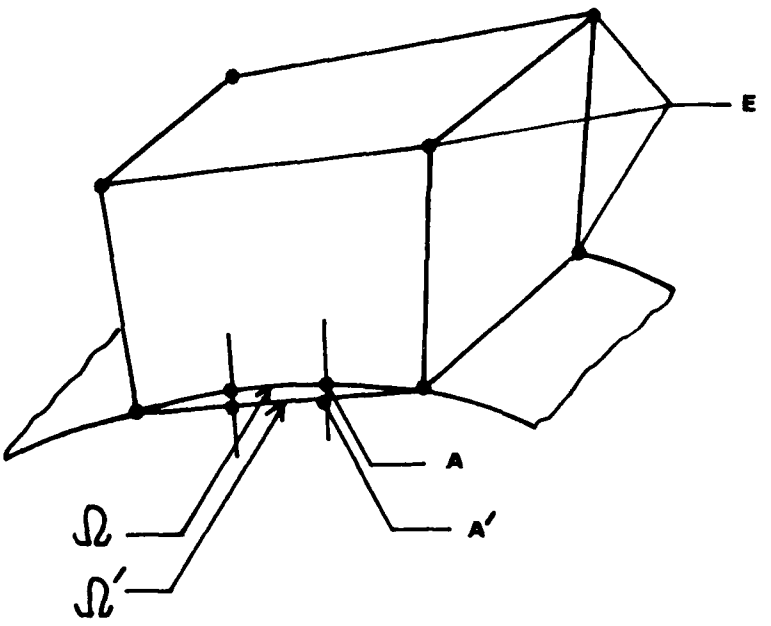


Fig. 3.11 Isoparametric surface definition for an element

3.4.2 Definition of Boundary Surfaces

The boundary surface definition related to the relocation process is an option in the grid generation program and is handled as a separate unit. Except for some built-in surface definitions, any particular surface should be defined in a local coordinate system in a way that y-coordinate of a point on the surface is a function of x-z coordinates. Then, a function sub-program can easily be associated with the main program.

In addition to the function, the program also assumes specific data which is required for definition of a surface and transformation between global and local coordinate system. For each type of surface, certain parameters are specified. For example a cylinder is defined by its radius and its length. The local coordinate system is defined relative to the global coordinate system, in a general manner, by means of three points. As shown in Fig. 3.12, the point A is the origin of the local system, the point B is an arbitrary point on the positive local. Z axis and the point C is another arbitrary point on local XZ plane. The local unit vectors can then be obtained with respect to the global system as follows:

$$\underline{k}_1 = \underline{AB}/|\underline{AB}|$$

$$\underline{j}_1 = \underline{AB} \times \underline{AC}/|\underline{AB} \times \underline{AC}| \quad \ldots\ldots \tag{3.4.1}$$

$$\underline{i}_1 = \underline{e}_y \times \underline{e}_y$$

The unit vectors are then expressed in the following way, l_i, m_i, n_i being direction cosines:

$$\underline{l}_i = l_1\underline{i} + m_1\underline{j} + n_1\underline{k}$$

$$\underline{j}_i = l_2\underline{i} + m_2\underline{j} + n_2\underline{k} \quad \ldots\ldots \tag{3.4.2}$$

$$\underline{k}_i = l_3\underline{i} + m_3\underline{j} + n_3\underline{k}$$

The transformation of coordinates from global to local or vice versa is initiated by a translation of origins to the same point. In the case of transformation from global to local, vector $\underline{P}_T$ shown in Fig. 3.13 can be written as

$$\underline{P}_T = \underline{P} - \underline{A} = x\underline{i} + y\underline{j} + z\underline{k} \tag{3.4.3}$$

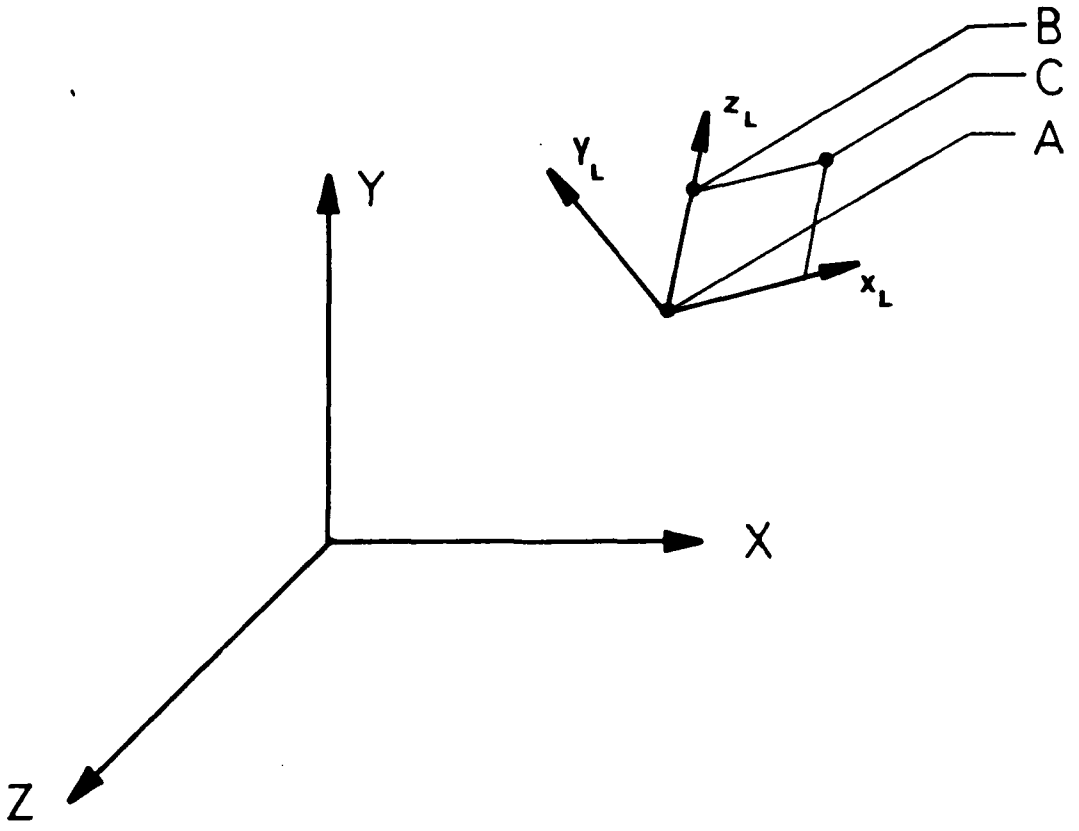


Fig. 3.12 Definition of local coordinate system for a boundary surface

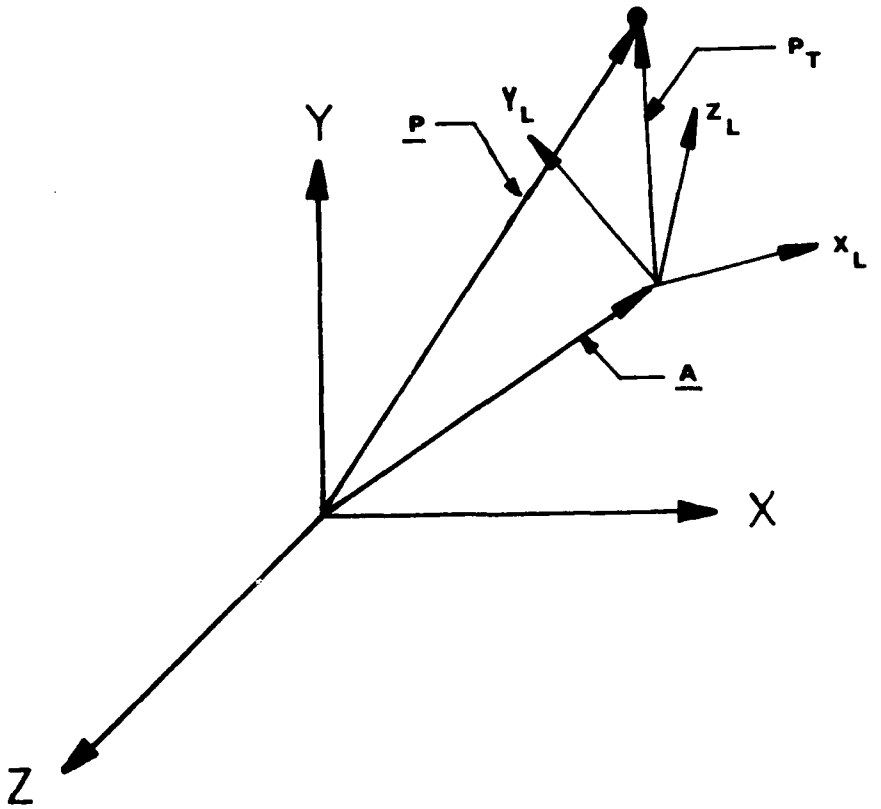


Fig. 3.13 Coordinate and vector transformation between local and global coordinate systems

It should be noted that $\underline{P}_T$ is still expressed in terms of global unit vectors and transformation is not complete until the translated global axis is rotated to obtain the local system. This can be obtained by a projection in the cartesian coordinate system where;

$$\text{if} \quad \underline{P}_T = x_1\underline{i} + y_1\underline{j} + z_1\underline{k} \quad ,$$

Then;
$$x_1 = yl_1 + ym_1 + zn_1 \tag{3.4.4a}$$

$$y_1 = xl_2 + ym_2 + zn_2$$

$$z_1 = xl_3 + ym_3 + zn_3$$

or in matrix form,

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.4.4b}$$

In the case of vector transformation, the translation step can be skipped, since the magnitude is normalized later.

The back transformation is obtained from the above relationship, this time by solving for x,y,z. Orthagonality of the transformation suggests that the inverse of the transformation matrix is merely its transpose.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \\ n_1 & n_2 & n_3 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{3.4.5}$$

It should be noted also, that,

$$\underline{P} = \underline{P}_T + \underline{A}$$

In the present study, the difinition of surfaces can be collected under two different types:

1. Geometric surfaces
2. Composite surfaces

Geometric surfaces are the ones which have a functional relationship valid all over the surface, like cylinders, spheres, ellipsoids of different diameters or some functional surfaces like a wing with a NACA 0012 profile. In addition to the local axis definition, the functions are specified in terms of x,y,z coordinates and some additional parameters.

On the other hand, composite surfaces are generally approximations of more complex surfaces. These are defined by interpolating polynomials between known surface points or surface sections. In this study, a composite surface definition for a wing surface which is defined at root and tip sections by discrete data points is presented. The basic approach to the problem is to fit a curve to the discrete data at sections and to interpolate linearly in between as shown in Fig. 3.14.

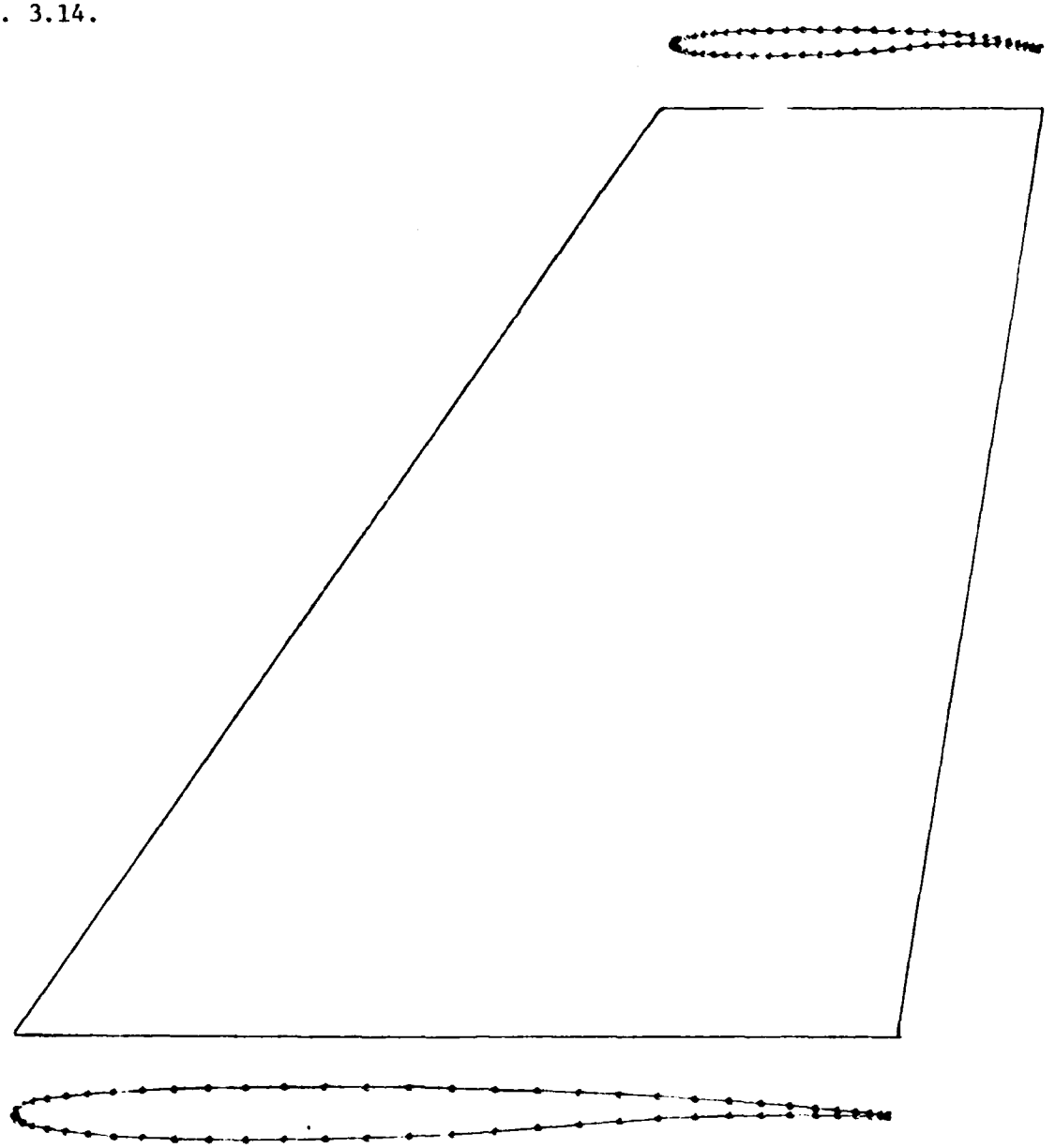


Fig. 3.14 Definition of a wing by using discrete data

In choosing the proper interpolating functions, the following basic requirements are to be considered:

a) the interpolating function itself and at least its first derivative has to be continuous at data points with no oscillatary tendencies; so that a smooth surface could be obtained,

b) the function has to be single valued which also brings up the necessity that upper and lower surfaces of the wing should be handled separately.

Since, considerable large data points are present, the Lagrangian interpolating polynomial of degree n passing through those (n+1) points is most likely to have undesirable oscillations. A composite curve, by fitting successive low-degree polynomials to successive groups of data points seems to avoid this problem but discontinuities of slopes at the functions become unacceptable.

The further possibility is to employ Hermitian interpolation functions, which interpolates on each interval $[x_i, x_{i+1}]$ by using higher-order polynomials while satisfying the continuity of the derivatives. A real valued function, f(x), which is defined by functional values and its derivatives at discrete points $x_1, x_2, \ldots, x_n$, in an interval [a,b], such that $a = x_1 < x_2 .. < x_n = b$, is interpolated by the piecewise-cubic polynomial function $P_i(x)$ as follows:

$$P_i(x) = C_{1,i} + C_{2,i}(x-x_i) + C_{3,i}(x-x_i)^2 + C_{4,i}(x-x_i)^3 \quad , \quad (i=1,.....n) \quad . \tag{3.4.6}$$

The coefficients of this polynomial are given, according to the Newtonian form of the interpolating polynomials [31] as follows:

$$\begin{aligned} C_{1,i} &= f_i = f(x_i) \\ C_{2,i} &= f_i' = f'(x_i) \\ C_{3,i} &= f[x_i, x_i, x_{i+1}] - f[x_i, x_i, x_{i+1}, x_{i+1}]\,(x_i - x_{i+1}) \\ C_{4,i} &= f[x_i, x_i, x_{i+1}, x_{i+1}] \end{aligned} \tag{3.4.7}$$

In terms of simple differencing, one can write,

$$C_{3,i} = \frac{f[x_i,x_{i+1}] - f_i}{\Delta x_i} - C_{4,i}\Delta x_i$$

$$C_{4,i} = \frac{f_{i+1} + f_i - 2f[x_i,x_{i+1}]}{(\Delta x_i)^2} \qquad (3.4.8)$$

where:

$$\Delta x_i = x_{i+1} - x_i$$

$$f[x_i,x_{i+1}] = \frac{f(x_{i+1}) - f(x_i)}{(x_{i+1} - x_i)}$$

But, in practice, it is often difficult to find the needed numbers of $f'(x_i)$. This condition suggests that a reasonable approximation to $f'(x)$ is necessary in the computation.

Piecewise cubic Bessel interpolation was chosen for this purpose to model airfoil problems. This function showed good agreement between the discretized data and interpolated results for tested wing profiles. In this case, the derivatives are approximated as shown below:

$$f'(x_i) = \frac{\Delta x_{i-1} f[x_i,x_{i+1}] + \Delta x_i f[x_{i-1},x_i]}{\Delta x_{i-1} + \Delta x_i} \qquad (3.4.9)$$

It should be noted that the above formulation requires the derivative values at the end points a and b also to be given as input data.

3.5 Relocation of the Boundary Nodes

Relocation of the boundary nodes is basically the result of the following two operations:

a) determination of the vector which is normal to the surface at the nodal point,

b) evaluation of the intersection point between normal vector and defined physical surface.

The unit normal vector for a surface is obtained as a result of cross-multiplication of two vectors, which are tangent to the surface at the nodal point coordinates. These tangent vectors denote the gradients of the position vector of the nodal point along two proper local coordinate directions. As it is seen in the Fig. 3.15, evaluation of local coordinate directions is based on the definition of the local system and the numbering the local surfaces accordingly.

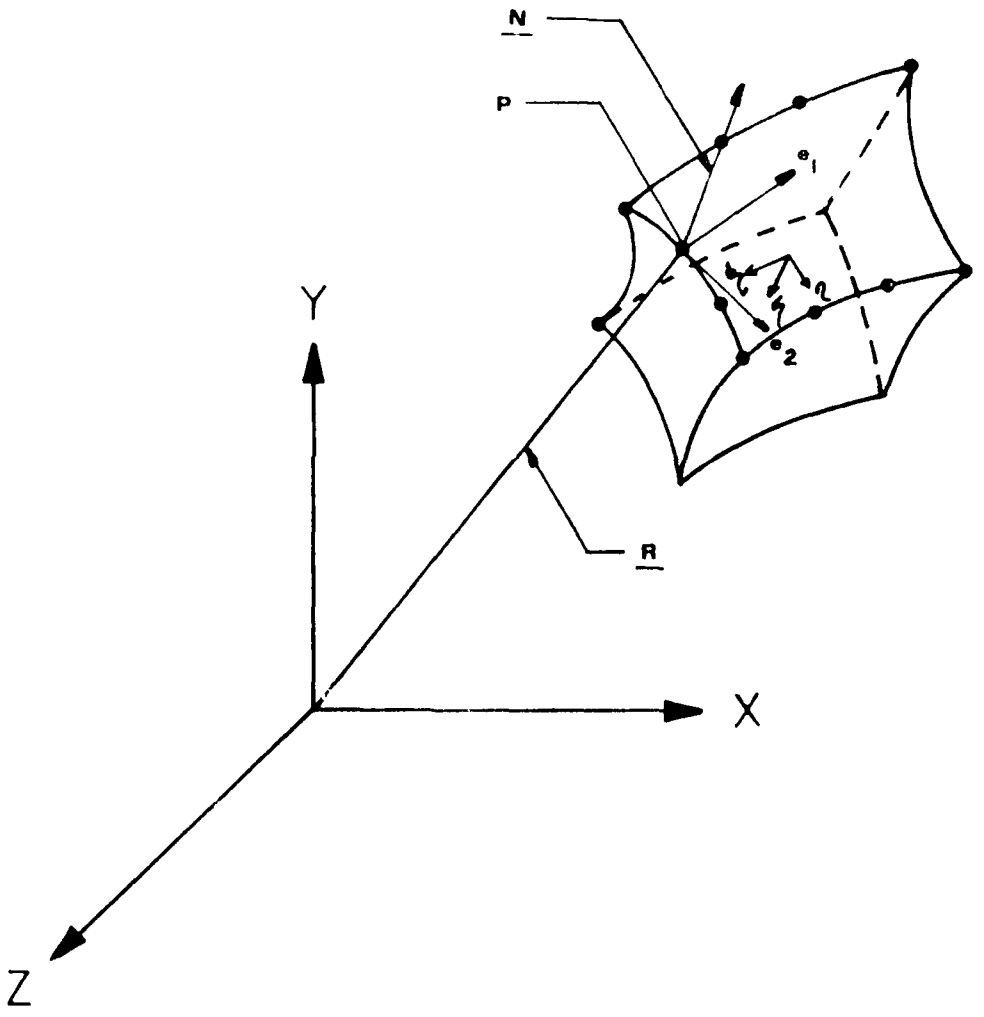


Fig. 3.15 Definition of a normal vector in an element

3.5.1 Evaluation of Surface Normal

In the above figures, the position vector to any point P on the surface of an element or block is given by:

$$\underline{R}(x,y,z) = R_x\underline{i} + R_y\underline{j} + R_z\underline{k} \tag{3.5.1}$$

For the specific point in the figure, the tangent vectors are obtained by taking directional derivatives of the position vector R along (η,ζ,ξ) directions. The program assumes the local number of the boundary surface to be given, so that local direction pairs along which the derivatives are calculated can be determined.

$$\underline{e}_1 = \frac{\partial \underline{R}}{\partial \xi} = \frac{\partial R_x}{\partial \xi}\underline{i} + \frac{\partial R_y}{\partial \xi}\underline{j} + \frac{\partial R_z}{\partial \xi}\underline{k}$$

$$\underline{e}_2 = \frac{\partial \underline{R}}{\partial \eta} = \text{etc.} \tag{3.5.2}$$

the normal then is:

$$\underline{N} = \frac{\underline{e}_1 \times \underline{e}_2}{|\underline{e}_1 \times \underline{e}_2|} \tag{3.5.3}$$

In order to evaluate the derivatives, the position vector components have to be expressed in terms of local coordinates which can be achieved by again using the serendipity type shape functions. The transformation between local and global point coordinates by using element nodal point coordinates is given as follows:

$$R_x = \sum_1^n N_i(\xi_p,\eta_p,\zeta_p)x_i$$

$$R_y = \sum_1^n N_i(\xi_p,\eta_p,\zeta_p)y_i \tag{3.5.4}$$

$$R_z = \sum_1^n N_i(\xi_p,\eta_p,\zeta_p)z_i$$

where n is the total number of nodal points in the element;

N_i = Nodal shape functions evaluated at the local coordinates ξ_p, η_p, ζ_p of the point P.

As a result, surface tangent vectors are expressed in terms of shape function derivatives and global nodal point coordinates as follows:

$$\underline{e}_1 = \sum_{1=1}^{n} \{(\frac{\partial N_i}{\partial \xi} x_i)\underline{i} + (\frac{\partial N_i}{\partial \xi} j_i)\underline{j} + (\frac{\partial N_i}{\partial \xi} z_i)\underline{k}\}$$

$$\underline{e}_2 = \text{etc.} \qquad (3.5.5)$$

3.5.2 Determination of the Point on the Physical Boundary Surface

The last step in the relocation process is the evaluation of the intersection point of the normal to the isoparametric surface and the defined boundary surface. The following Fig. 3.16 shows the process for a typical case.

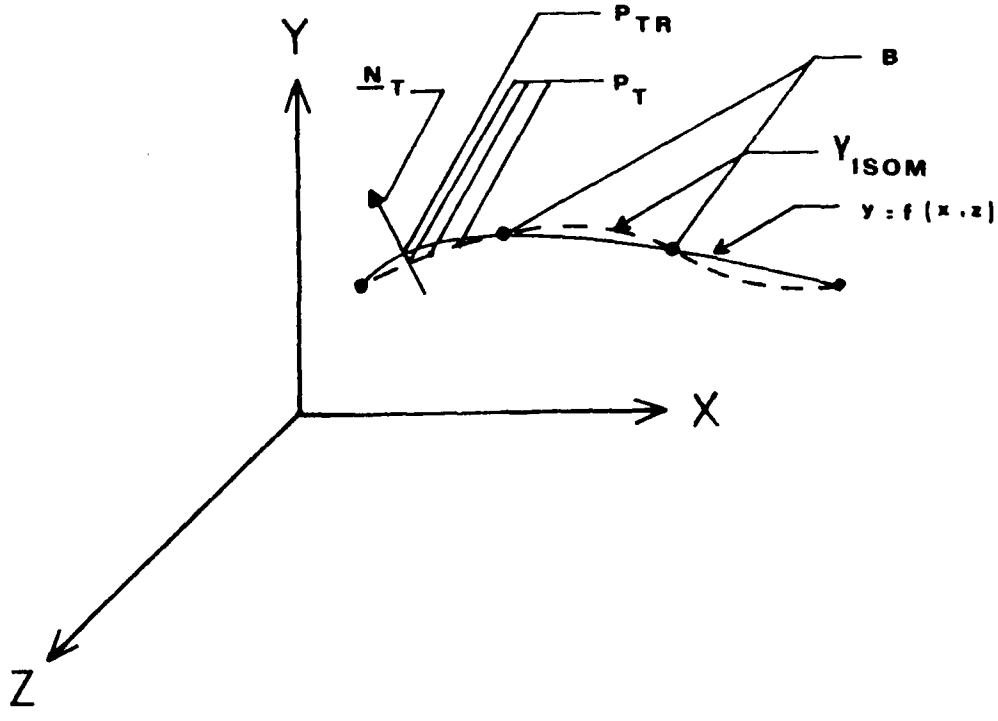


Fig. 3.16 The relocation process

Points P are the generated nodal points inside the block which do not lie necessarily on the physical surface, $\underline{N}_T$ is the normal vector transformed into locally defined boundary surface coordinate system and y is the boundary surface defined with a functional relationship. Since boundary surfaces are defined locally, associated with a transformation matrix, the point P and the vector N can be transformed into P_T and N_T in local coordinates quite early. P_{TR} is the relocated point in the local coordinate system.

Evaluation of the intersection point is an iterative process and can be viewed as a three-dimensional application of the method of successive approximations except the convergence is carried out along the normal vector.

Geometric representation of the process in two-dimensions as shown in Fig. 3.17, where,

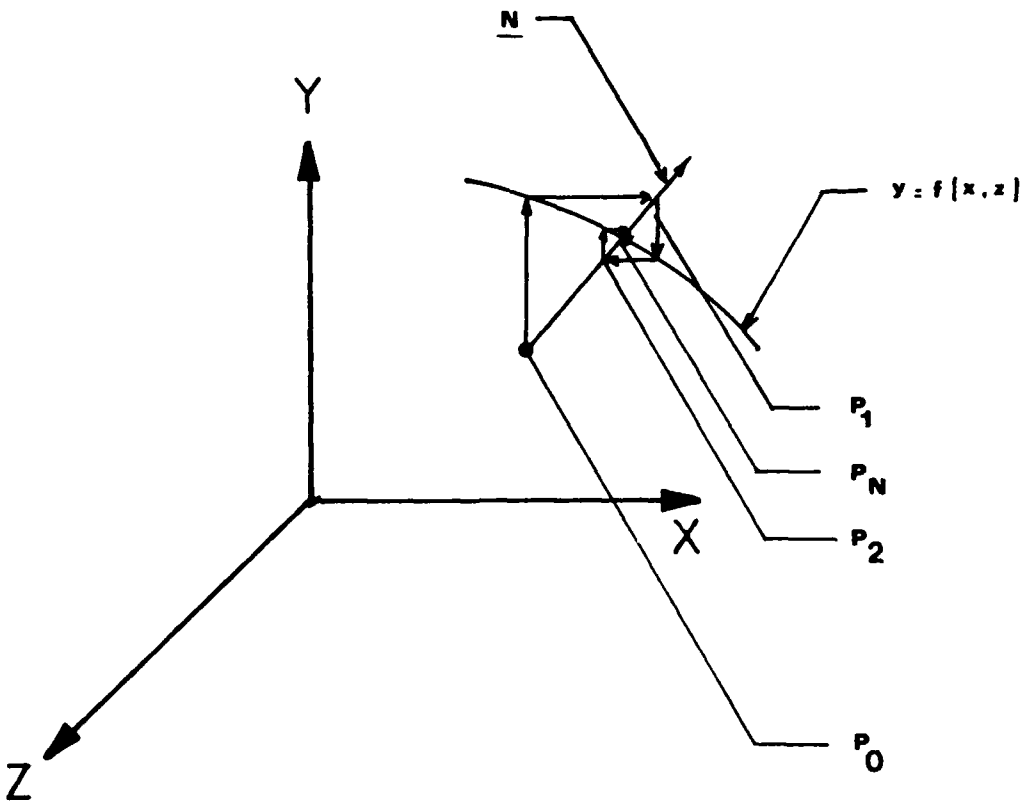


Fig. 3.17 The method of successive approximations

$$\underline{N} = l\underline{i} + m\underline{j} + n\underline{k}$$

The iterative procedure is summarized in equation (3.5.6)

$$\underline{P}_n = x_n\underline{i} + y_n\underline{j} + z_n\underline{k} \qquad (n=1,i)$$

Step 1 $\quad y_{n+1} = y(x_n, z_n)$

Step 2 $\quad \lambda_n = (y_{n+1} - y_o)/m$ (3.5.6)

Step 3 $\quad \underline{P}_{n+1} = \underline{P}^o + \lambda_n\underline{N} = y_{n+1}\underline{i} + y_{n+1}\underline{j} + z_{n+1}\underline{k}$

Go back to step 1.

The above iterative procedure continues until desired convergence which is the percent change in λ is obtained. Unless the angle between the normal vector and surface tangents becomes too small, the convergence is guaranteed that P_n will get closer and closer to the solution P_{TR}.

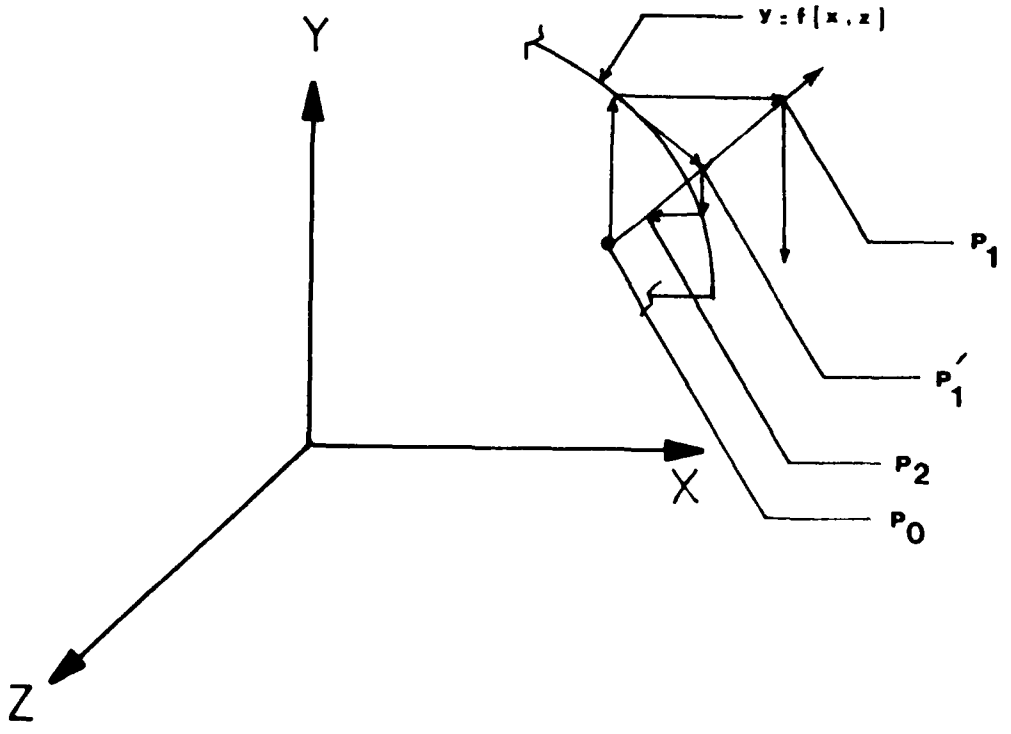


Fig. 3.18 Ill-conditioned successive approximations

Sometimes a point on the surface can not be determined by this scheme, this ill-conditioned situation shown above, in Fig. 3.18, can be improved by introducing a relaxation factor ω on λ_η ,

$$\lambda_\nu = \omega(y_{n+1} - y_o)/m \quad . \tag{3.5.7}$$

If the new point P_{n+1} is out of range of surface, the relaxation factor which is initially one, is divided by two. This division is repeated until y_{n+1} can be defined on the surface, eventually, a point P_1' can be computed for which this ill-conditioned situation does not occur.

The relocation process applied to the NACA0012 airfoil profile is shown in Fig. 3.19 and 3.20a,b. The first figure shows the cubic control points on the curved edge along the streamwise direction of the airfoil. THe generated surface by isoparametric transformation is shown in Fig. 3.20a. It is obvious that the smaller curvature at the leading edge compared to the rest of the surface is lost due to lack of information around this region. After the relaxation is applied all over the surface, the improvement of the leading edge points are clearly observed in Fig. 3.20b.

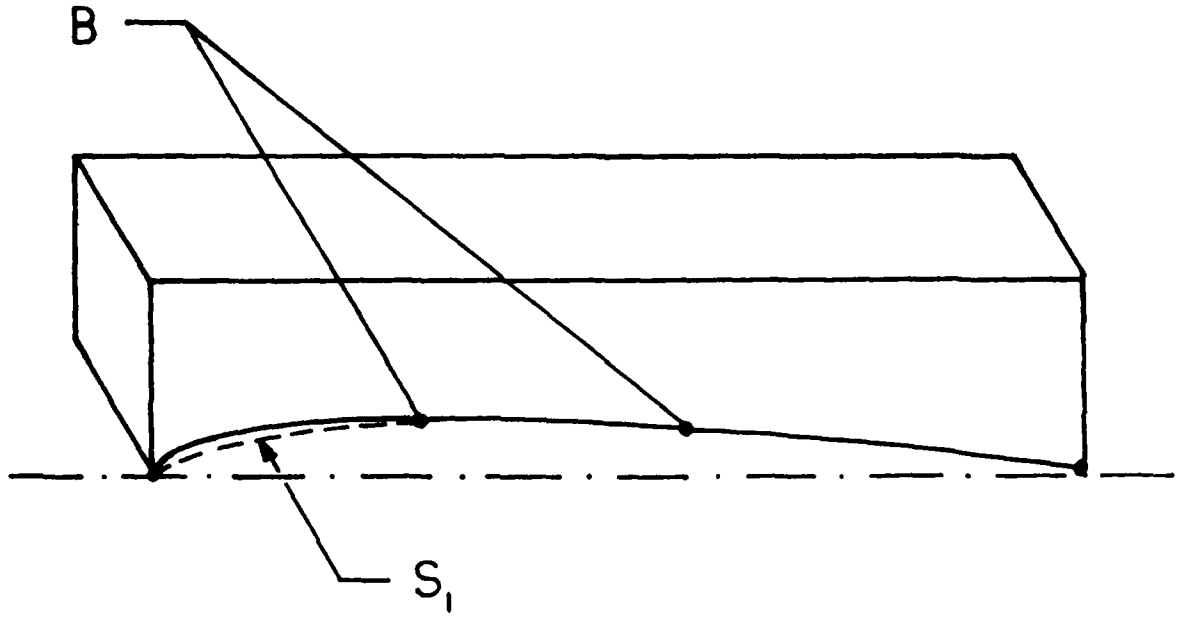


Fig. 3.19 Definition of the cubic block edge for NACA0012 airfoil profile

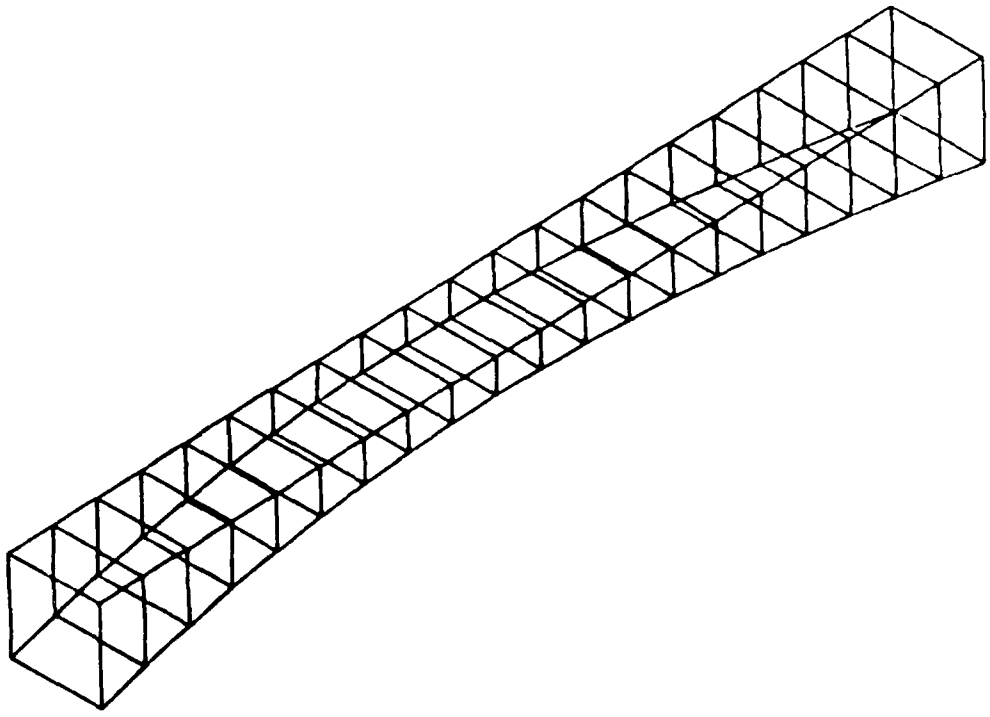


Fig. 3.20a Grid distribution using isoparametric mapping over NACA0012 airfoil

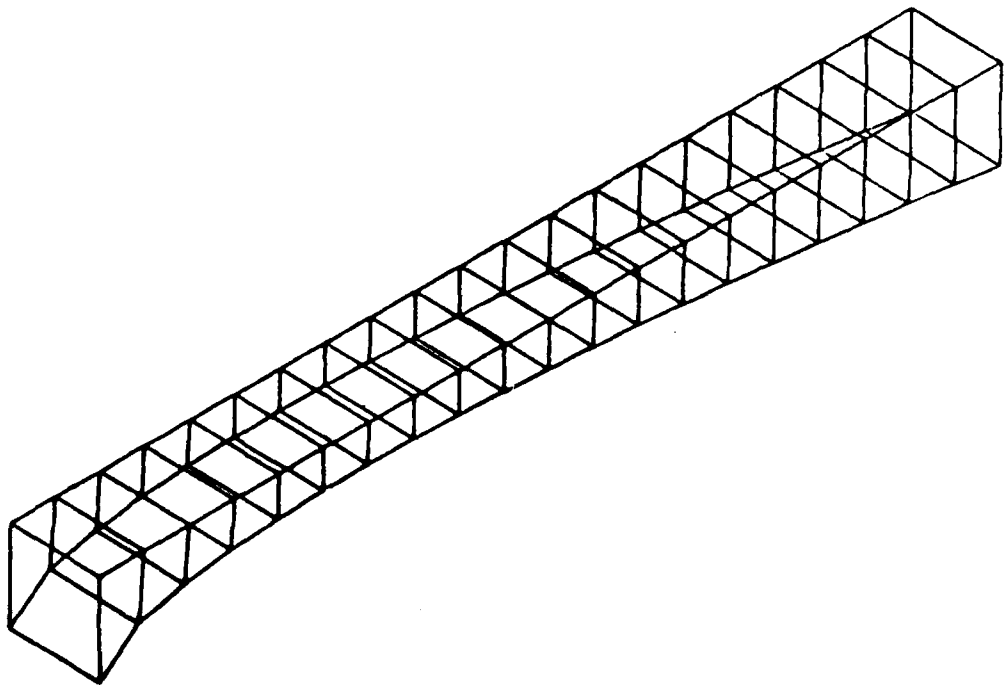


Fig. 3.20b Modified grid distribution after relocation of surface nodal points over NACA0012 airfoil

IV. DESIGN OF A FINITE ELEMENT GRID

4.1 DESCRIPTION OF THE TEST PROBLEM

To study the applicability of the developed numerical scheme, the transonic flow around a wing-body combination was analyzed. This particular wing was designed at Lockheed-Georgia wind-tunnel by Hinson and Burges [32]. With an aspect ratio of 3.8, it was originally designed for transonic cruise and tested in the presence of a body in a high Reynolds number wind-tunnel. The geometry of the wing is defined at the root and tip sections by discrete data points. Table 4.1 gives the general wing characteristics. The measurements are non-dimensionalized by the cord length at the root as shown in Table 4.2. The wing coordinates at other spanwise stations are then evaluated by linear interpolation between the root and the tip sections.

The fuselage used in the test is a simple shape of an elliptical forebody and afterbody with a constant section in the wing region. In the present study, the fuselage was taken as an infinite cylinder with a constant cross-section.

The overall geometric characteristics of the wing model is summarized in Fig. 4.1.

4.2 GENERATION OF THE COMPUTATIONAL GRID

In the present study, the computational domain is represented by a finite element grid generated by the scheme described in the previous chapter. The physical space consists of an infinite fuselage having a circular cross-section of constant radius and the wing attached to the fuselage. The computational space is truncated at finite distances from the wing surface. It is assumed that the flow is symmetric about the vertical plane containing the fuselage center line, so that symmetry conditions can be applied and the flow has to be analyzed only in the half space.

For results presented here, the far-field boundaries are placed approximately three chord-lengths from the wing surface in the streamwise and surface normal directions; in the spanwise direction, far-field is

Table 4.1 Wing characteristics

AR	3.8
λ	0.4
$\Lambda_{c/4}$, deg.	30.0
θ_r, deg.	2.50
θ_t, deg.	-4.00
$(t/c)_r$, %	6.0
$(t/c)_t$, %	6.0
S/2, cm^2 (in^2)	530.0 (82.1)
b/2, cm (in.)	31.8 (12.5)
C_r, cm (in.)	23.88 (9.40)
C_t, cm (in.)	9.55 (3.76)
MAC, cm (in.)	17.71 (6.974)
Y_{MAC}, cm (in.)	13.60 (5.355)

Table 4.2 Wing geometry

	ROOT SECTION		TIP SECTION	
X/C	Z_U/C	Z_L/C	Z_U/C	Z_L/C
.00000	.00000	.00000	.00000	.00000
.00241	.00617	-.00528	.00507	-.00606
.00961	.01181	-.00895	.00972	-.01066
.02153	.01649	-.01198	.01401	-.01408
.03806	.01991	-.01511	.01770	-.01691
.05904	.02268	-.01839	.02110	-.01951
.08427	.02517	-.02111	.02421	-.02161
.11349	.02737	-.02333	.02700	-.02325
.14645	.02925	-.02503	.02949	-.02439
.18280	.03075	-.02618	.03168	-.02492
.22221	.03191	-.02691	.03360	-.02498
.26430	.03277	-.02705	.03522	-.02446
.30866	.03330	-.02669	.03654	-.02344
.35486	.03346	-.02582	.03762	-.02180
.40245	.03325	-.02458	.03847	-.01967
.45099	.03258	-.02287	.03905	-.01689
.50000	.03155	-.02070	.03933	-.01361
.54901	.03013	-.01768	.03922	-.00950
.59755	.02842	-.01376	.03882	-.00396
.64514	.02639	-.00985	.03799	.00042
.69134	.02417	-.00615	.03669	.00474
.73570	.02178	-.00316	.03491	.00814
.77779	.01925	-.00109	.03258	.01020
.81720	.01660	.00003	.02962	.01087
.85355	.01388	.00043	.02608	.01026
.88651	.01116	.00043	.02211	.00867
.91573	.00865	.00032	.01793	.00651
.94096	.00644	.00012	.01379	.00417
.96194	.00459	-.00021	.00991	.00196
.97847	.00308	-.00055	.00674	.00056
.99039	.00196	-.00082	.00445	-.00138
.99759	.00130	-.00102	.00305	-.00227
1.00000	.00109	-.00109	.00259	-.00257

located one span-length from the wing-tip [14,9]. The grid is chosen to be of C-type which wraps around the wing leading edge and becomes rectangular grid past the wing trailing edge. The elements generated in this way are proved to be the most suitable for modelling the sharp gradients of flow variables and matching the flow conditions.

4.2.1 General Considerations and Block Definitions

In the numerical solution of a particular fluid mechanics problem, the accuracy of the solution and the computation time strongly depend on the computational grid employed in the analysis. Therefore, the flexibility and the limitations of a grid generation scheme is important in obtaining accurate and efficient solutions.

In general, sub-regions of flow domain where high gradients of flow parameters are expected, the grid has to be finer than other regions to maintain the same level of accuracy. It is also desirable to maintain other properties like the orthogonality of the grid and streamlining to the flow direction to reduce artificial viscosity effects.

In the present transonic flow analysis around wing-body combinations, the wing has a sharp leading edge. Around this region high gradients of velocity were measured. In addition, the physical boundary around the leading edge shows a rapid change in the curvature which requires considerable concentration of grid points, in order to represent the exact boundaries without losing the details of the leading edge.

Considering the above physical characteristics of the flow problem and the limitations of the developed grid generation scheme such as:

a) physical surfaces can only be introduced as block surfaces, not inside the blocks,

b) a surface of a block can represent at most a third-degree polynomial surface with a smooth curvature change,

c) only a single gradient value can be defined along an edge of a block,

the computational grid shown in Fig. 4.2a,b,c was designed. The wing and vortex sheet is placed in between two layers of blocks in surface-normal direction by exploiting the slit option of the scheme. In the streamline direction, three layers of blocks are employed by having the first blocks coupled to each other in the flow direction. With this combination, a

C-type radial grid can be obtained and high gradients of elements around the leading edge can be introduced. In the spanwise direction, physical constraints require 4 block layers to be used. The block layer after the wing performs the transition between spanwise far-field and the wing tip. Fig. 4.3 shows the proposed block structure in the upper side of the wing and body surface, which is composed of 24 blocks in the whole domain. The block definitions in the global and natural coordinate system are shown in Fig. 4.4 and 4.5 respectively.

The entire block structure which is the combined form of block layers described above is presented in Fig. 4.6. It should be noted that although all the block definition points are connected to each other by linear lines, as it will be seen later, the generation of elements inside the blocks is based on the higher-order interpolation functions along the edges of each block.

4.2.2 The Grid Distribution

The element distribution inside the blocks is determined by the number of elements along three principal directions and the gradients defined along the block edges. Although these specifications are separately performed for each block, grid generation scheme assumes continuation of the same number of elements inside the blocks which are connected to each other along the principal directions. Also, it is required that the gradient definition for the same edge shared by up to four blocks should be the same. Fig. 4.7a,b,c,d show an arbitrary element distribution inside blocks.

In the present analysis, since the accuracy of the solution on the wing surface is of major importance, the first layer elements on the surfaces is kept close to the surface. It should be remembered that for eight noded bi-linear elements, centroidal values of these elements represent the flow on the surface. The element distribution around the leading edge is designed to be fine along the streamwise direction while over the wing a smooth change towards a coarser grid is aimed for better accuracy and efficiency. The farther the elements are from wing-body surfaces, the longer they become, since the flow variables do not vary rapidly at the far-field.

Figures 4.8 to 4.10 show the grid distribution along streamwise and spanwise directions at typical sections. The grid which is also employed for the flow analysis has 28 elements in streamwise direction; 18 of them

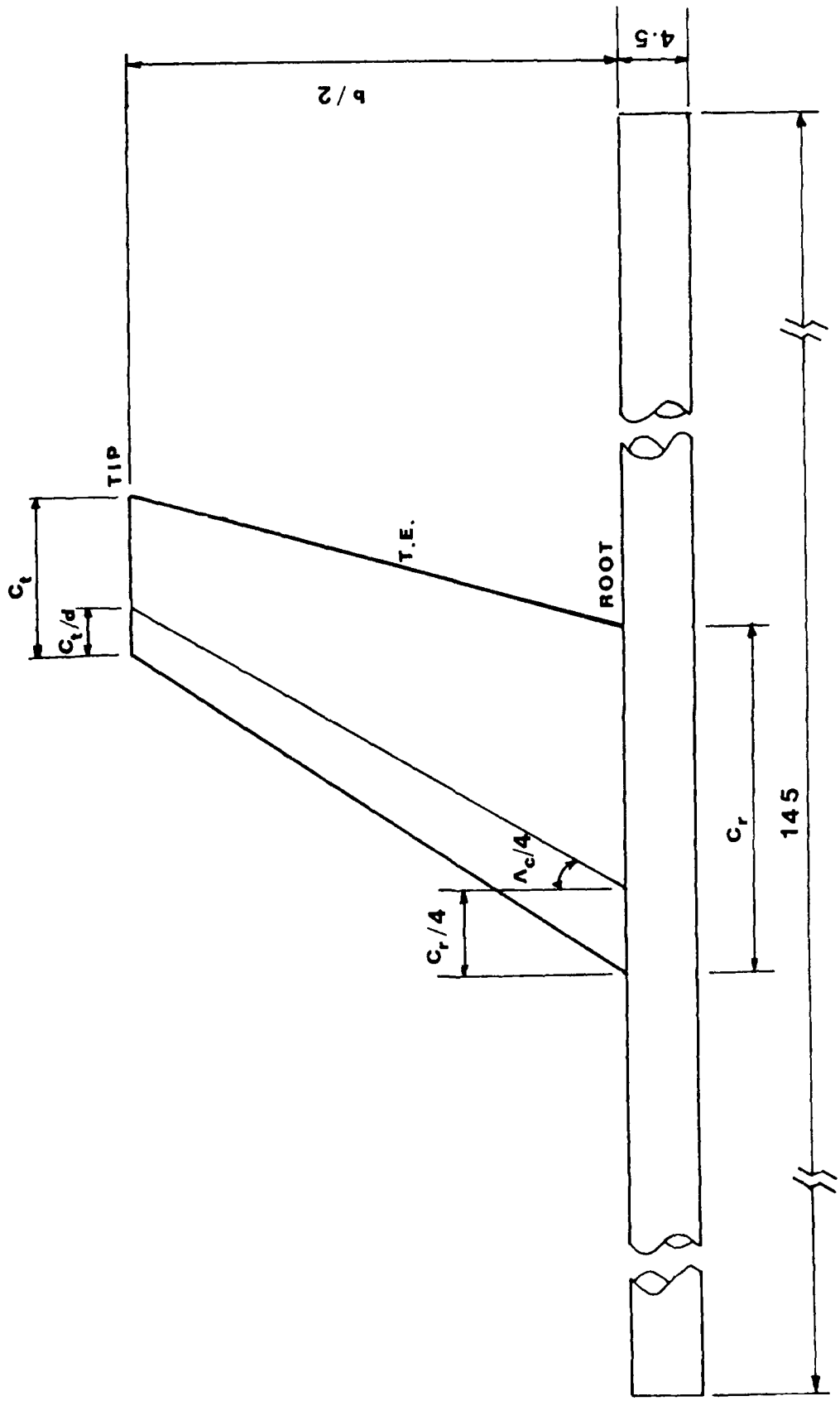


Fig. 4.1 The wing-body geometry

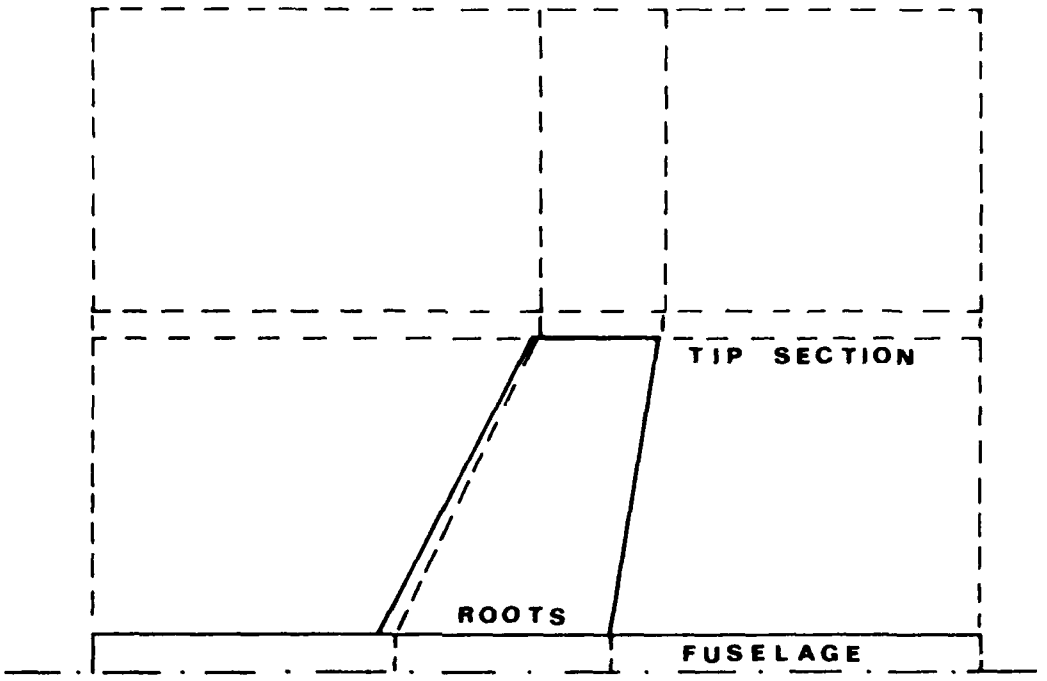


Fig. 4.2a Description of blocks, I-K section

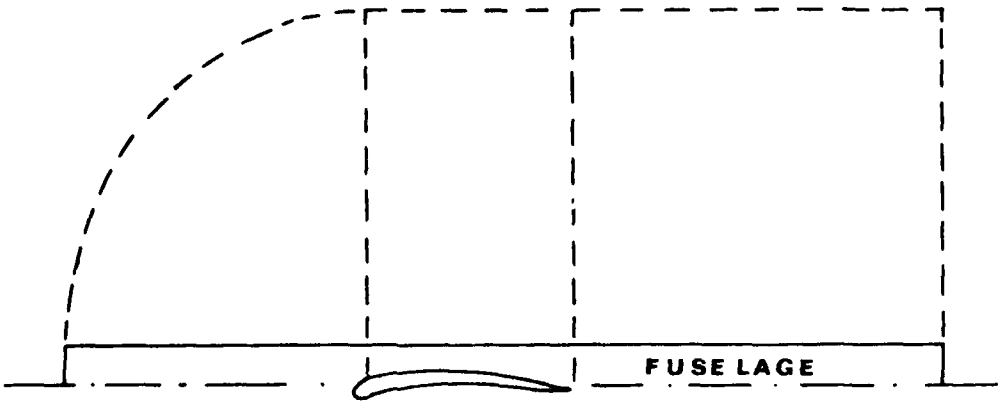


Fig. 4.2b Description of blocks, I-J section

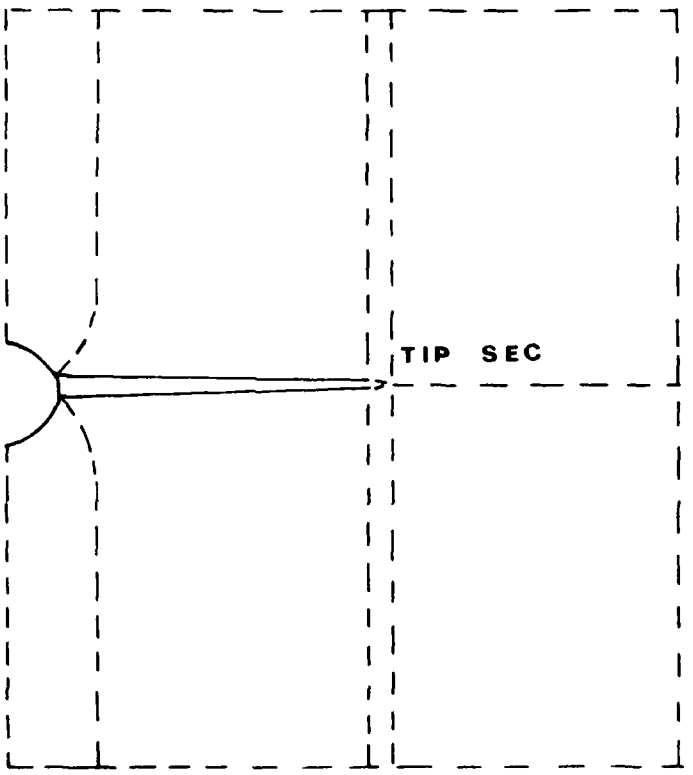


Fig. 4.2c Description of blocks, J-K section

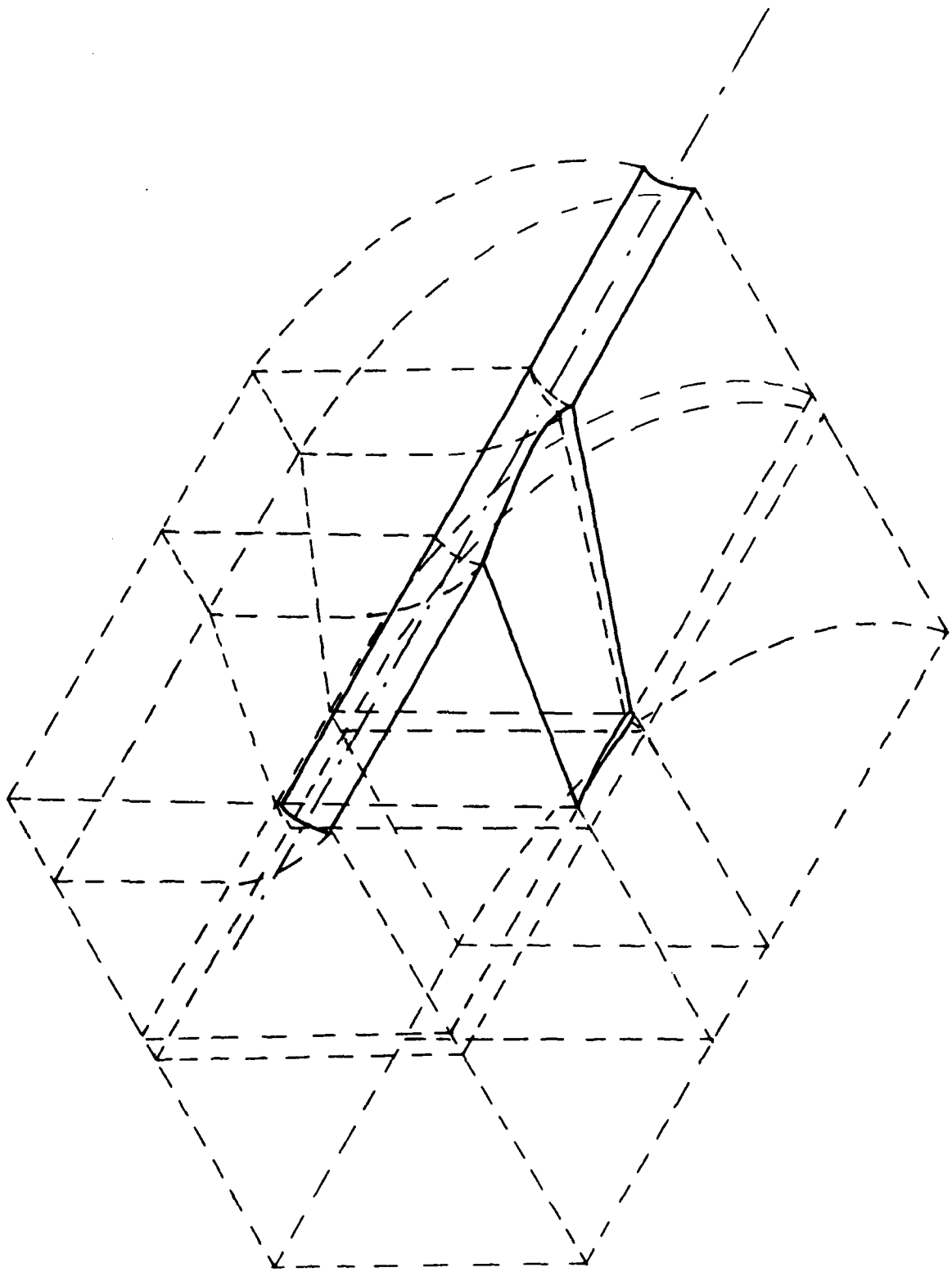


Fig. 4.3 The proposed block structure

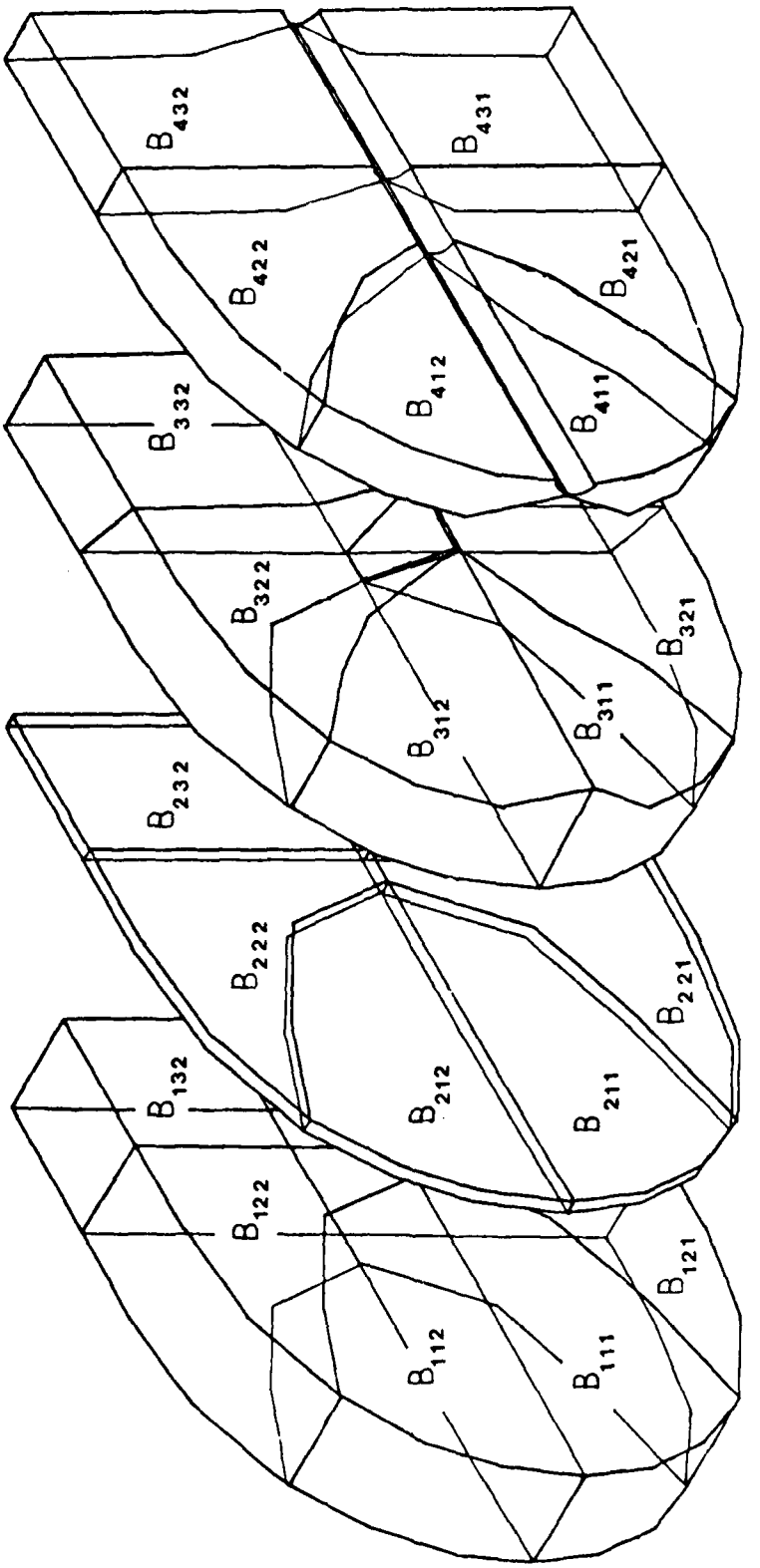


Fig. 4.4 The separate block layers

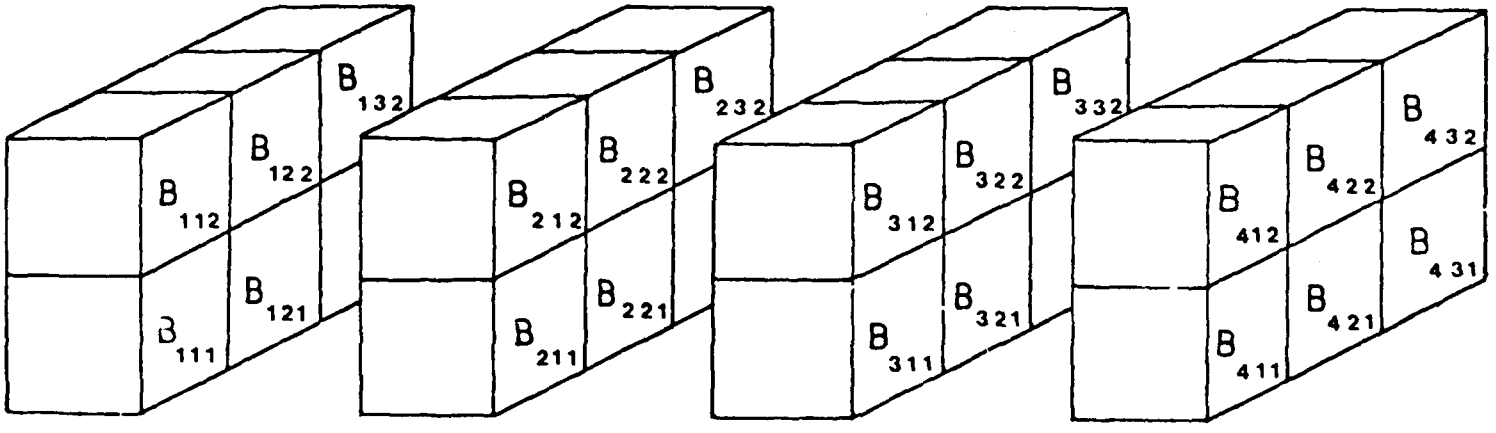


Fig. 4.5 The block structure mapped into natural coordinate system, I, J, K locations of blocks are denoted

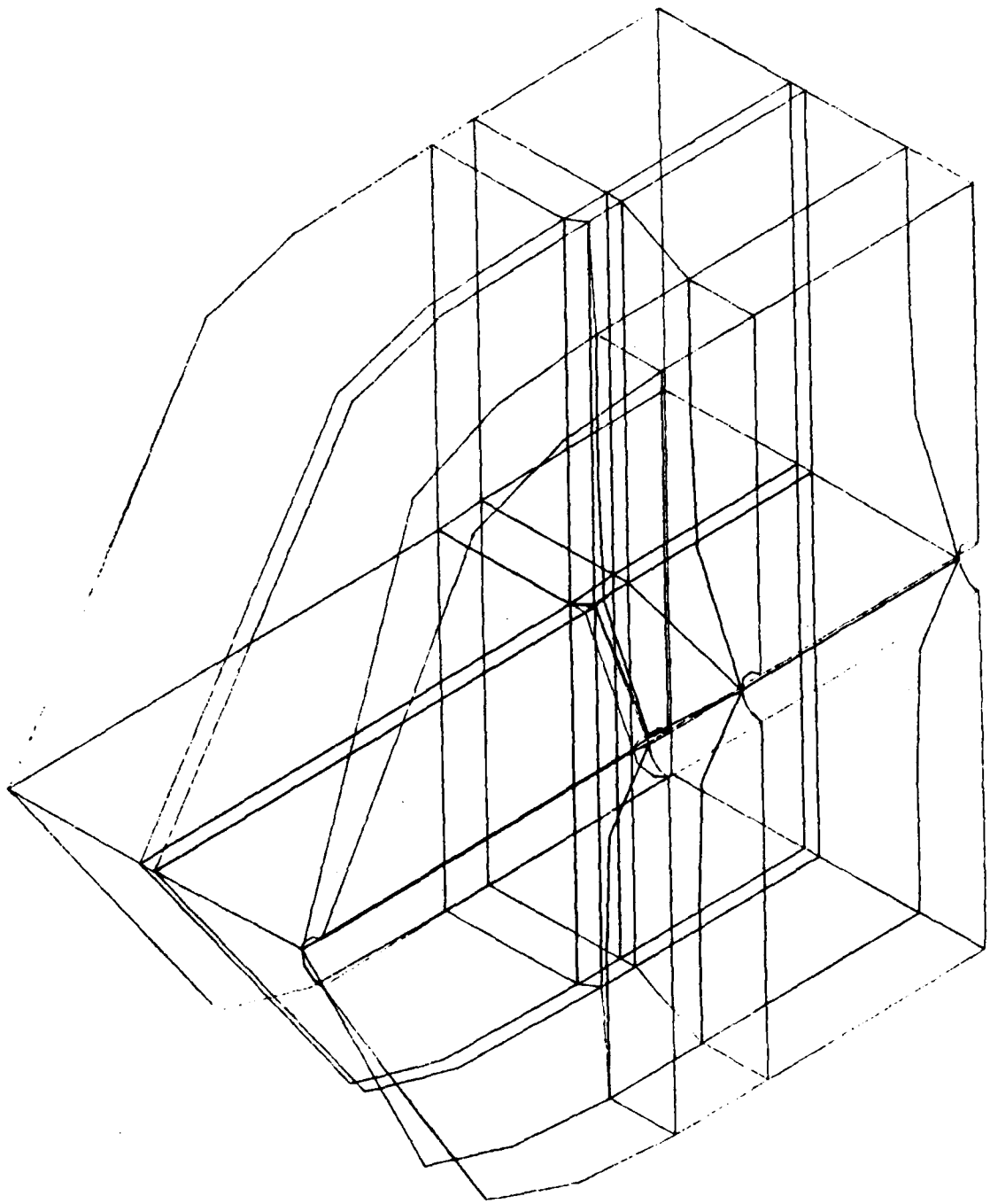


Fig. 4.6 The total block structure

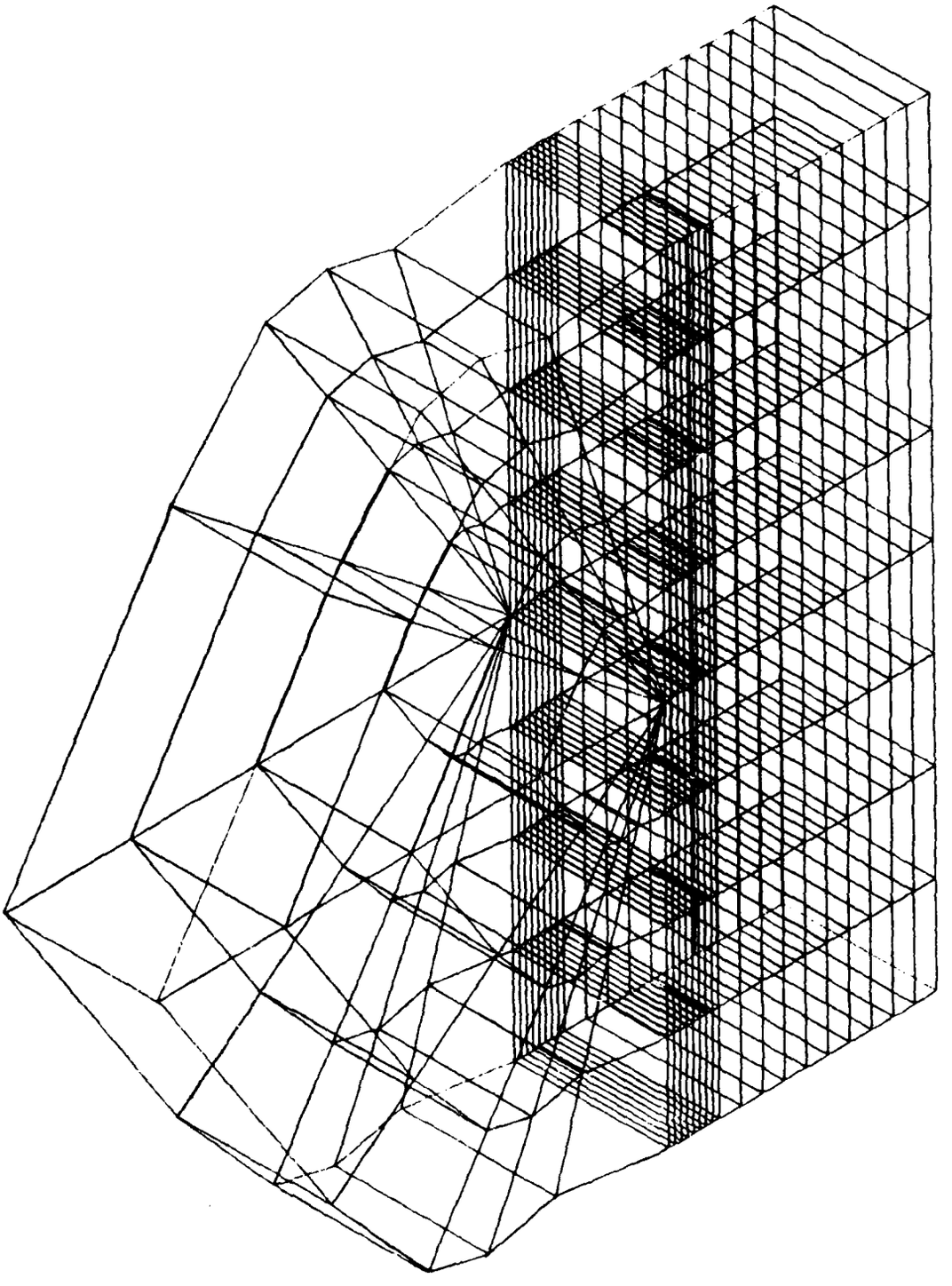


Fig. 4.7a An arbitrary element distribution in the block at the far-field

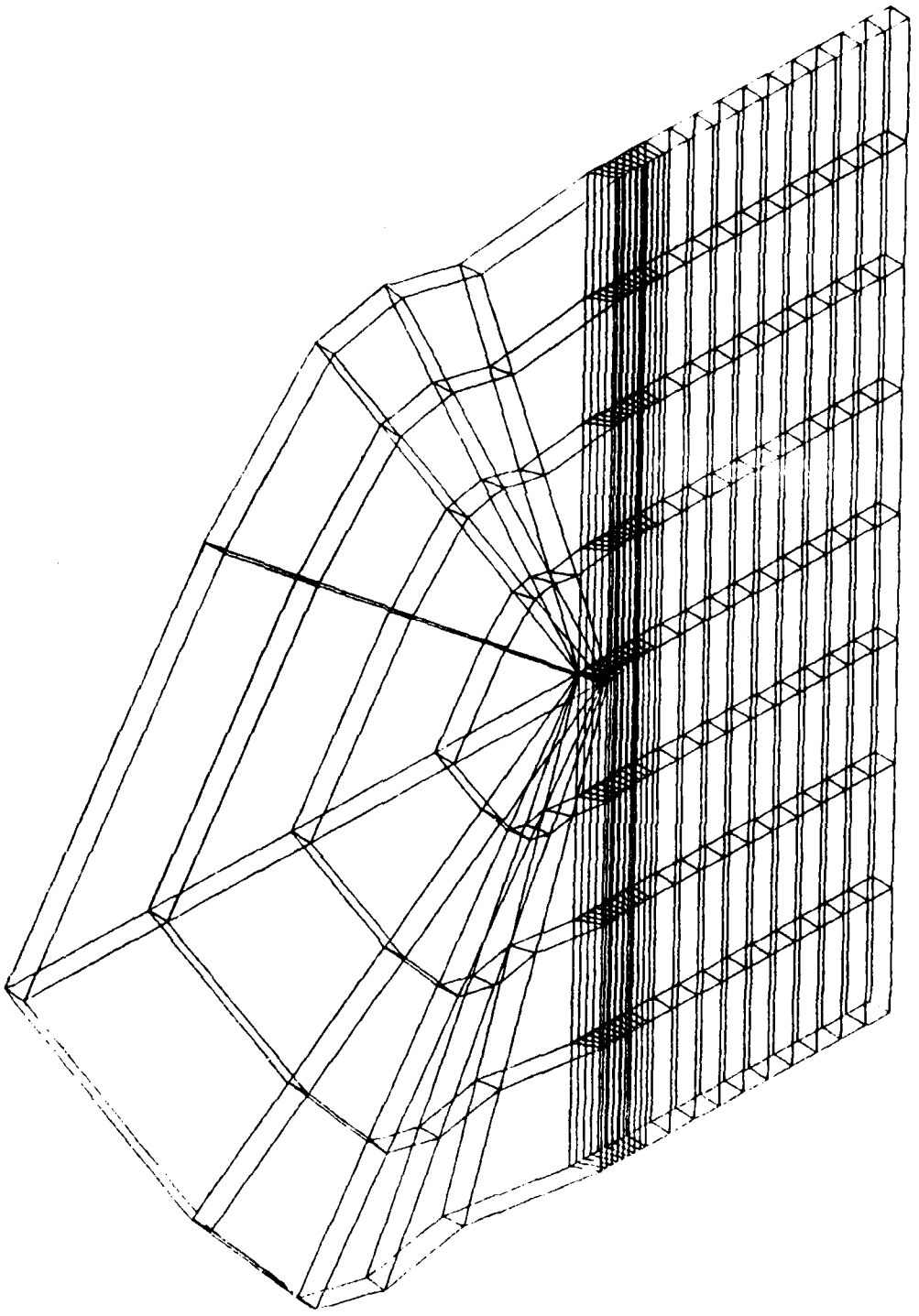


Fig. 4.7b An arbitrary element distribution in the block at the transition region

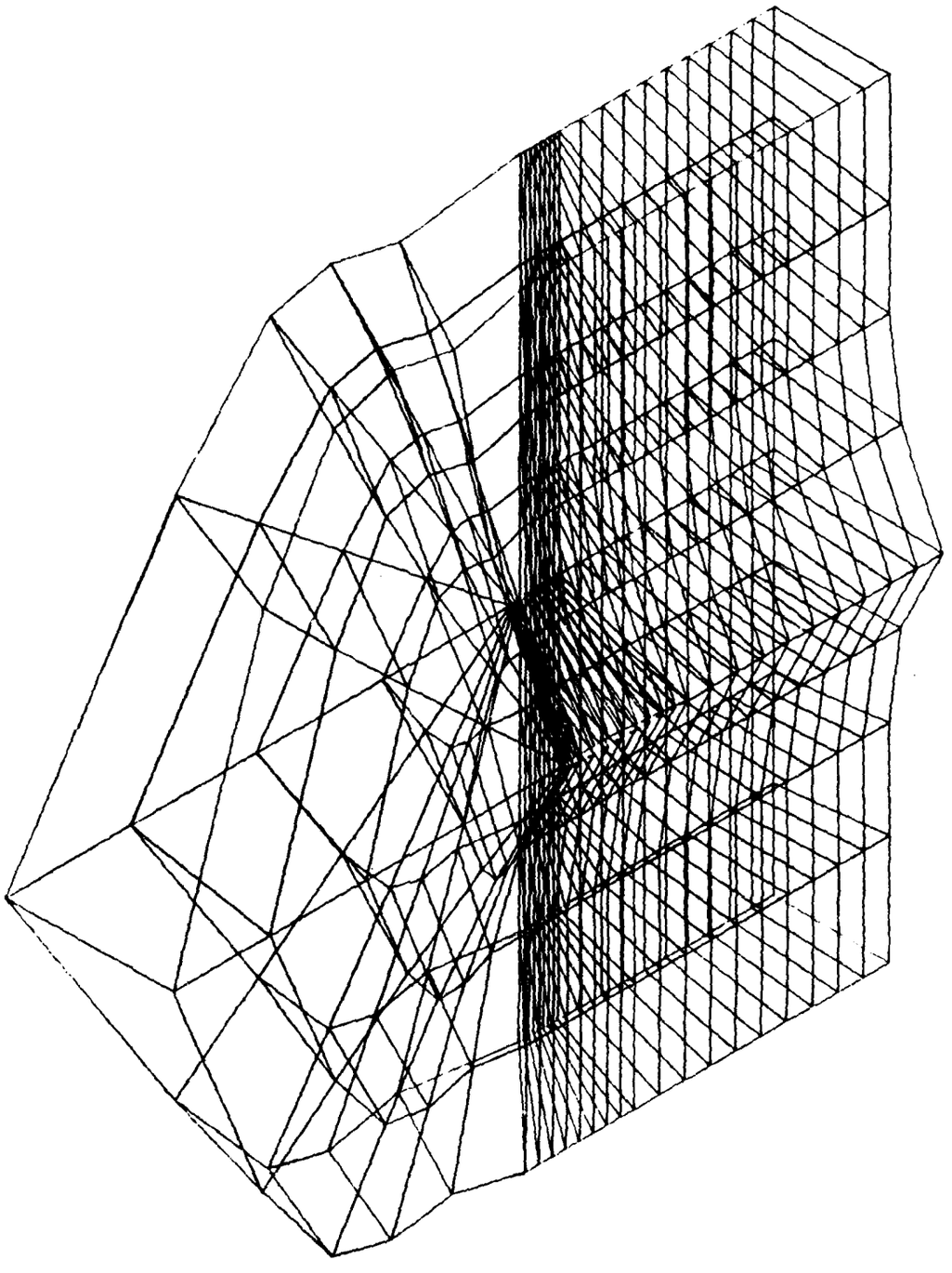


Fig. 4.7c An arbitrary element distribution in the block over the wing

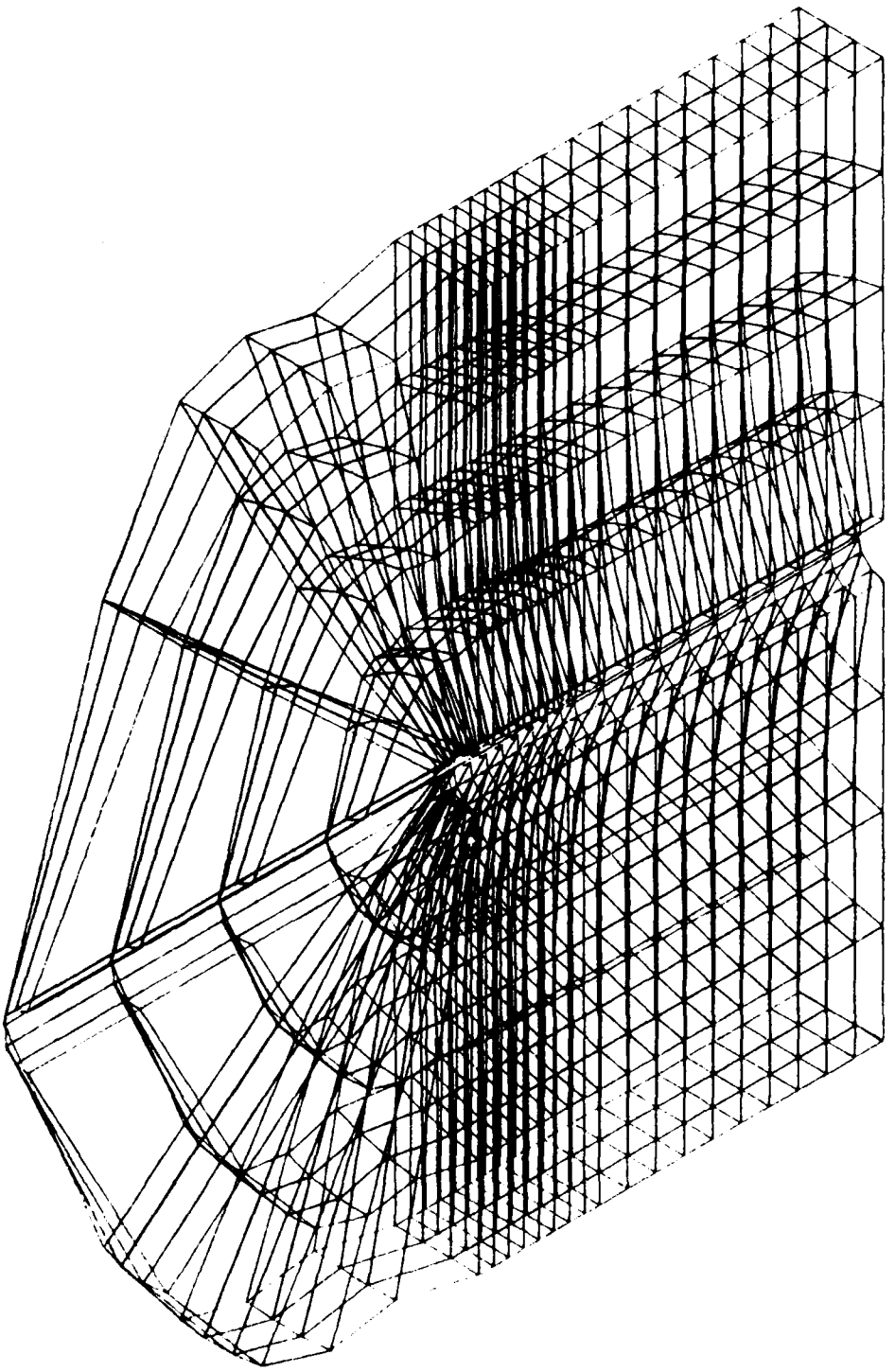


Fig. 4.7d An arbitrary element distribution in the block over the body

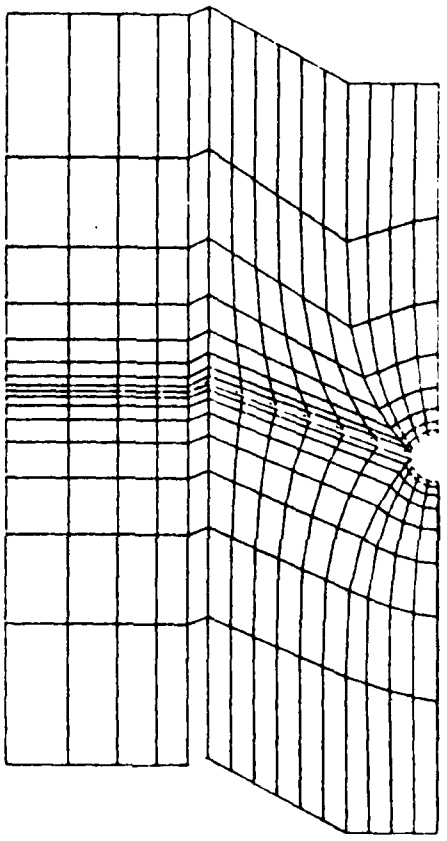


Fig. 4.8 Grid distribution, J-K section over the wing-body

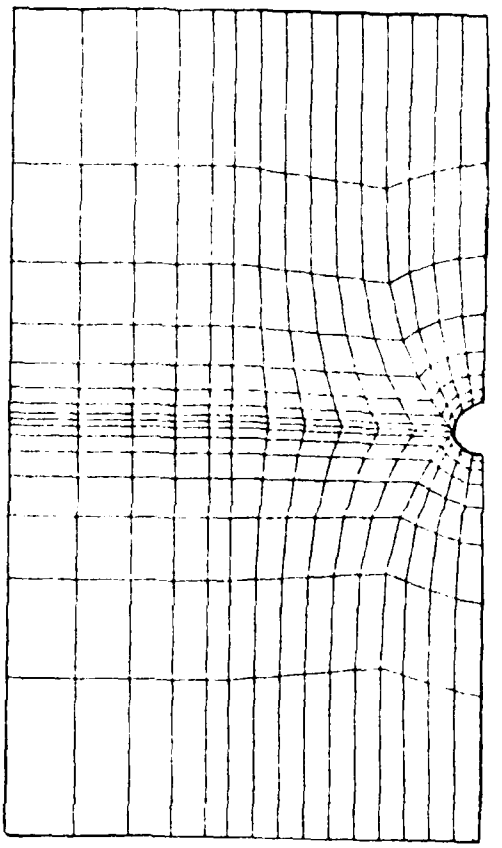


Fig. 4.9 Grid distribution, J-K section over the downstream far-field

located on the wing. There are 15 elements in spanwise direction; 4 of them on the body and 6 of them on the wing. In the third direction, normal to the wing upper and lower surfaces, 16 elements are utilized with the wing located at the middle. The total number of elements adds up to 6720. As it can be seen in Fig. 4.10, in each block, element gradients along streamwise direction are assumed to be uniform; i.e. each element inside the block has the same length in the streamwise direction. In the direction normal to the wing surface, gradients are specified in such a manner that the ratio of the element size on the surface that of the one at the far-field is 1/25. The peculiar shape of the radiality at the upstream boundary shown in Fig. 4.10 is the result of the shifted higher-order block edge nodes from correct 1/3 locations as discussed in section (3.2.1). This shift was made to reduce the distortion of the elements around the leading edge. The Fig. 4.11 shows the first layer of elements over the upper surface of the wing and body. Here, the collapsed block edge and the triangular wedge type elements over the body can also be seen clearly. Figure 4.12 and 4.13 are examples of element distribution inside blocks, in particular, blocks around leading edge bottom surface. The final grid distribution over the wing-body combination, including only the surface nodes of the elements, is shown in Fig. 4.14.

As it was mentioned before, in order to resolve rapid changes in the flow variables, higher-order elements have been employed at the critical flow regions. For this particular grid, two cubic element layers were placed over each of the wing upper and lower surfaces. Each layer had 15 cubic elements along the streamwise direction and 6 in the spanwise direction. The total number of cubic elements in the grid was 360.

It should be noted that quite a small amount of input data was required in order to supply such an element distribution. This is very important in the design of finite element girds from flexibility and efficiency points of view.

4.3 GRID STUDIES AND RESULTS

4.3.1 Case I. Basic Grid

The computational grid, designed in the pre ous section, was employed to anaiyze the transonic flow around the particular wing-body combination as an initial test case.

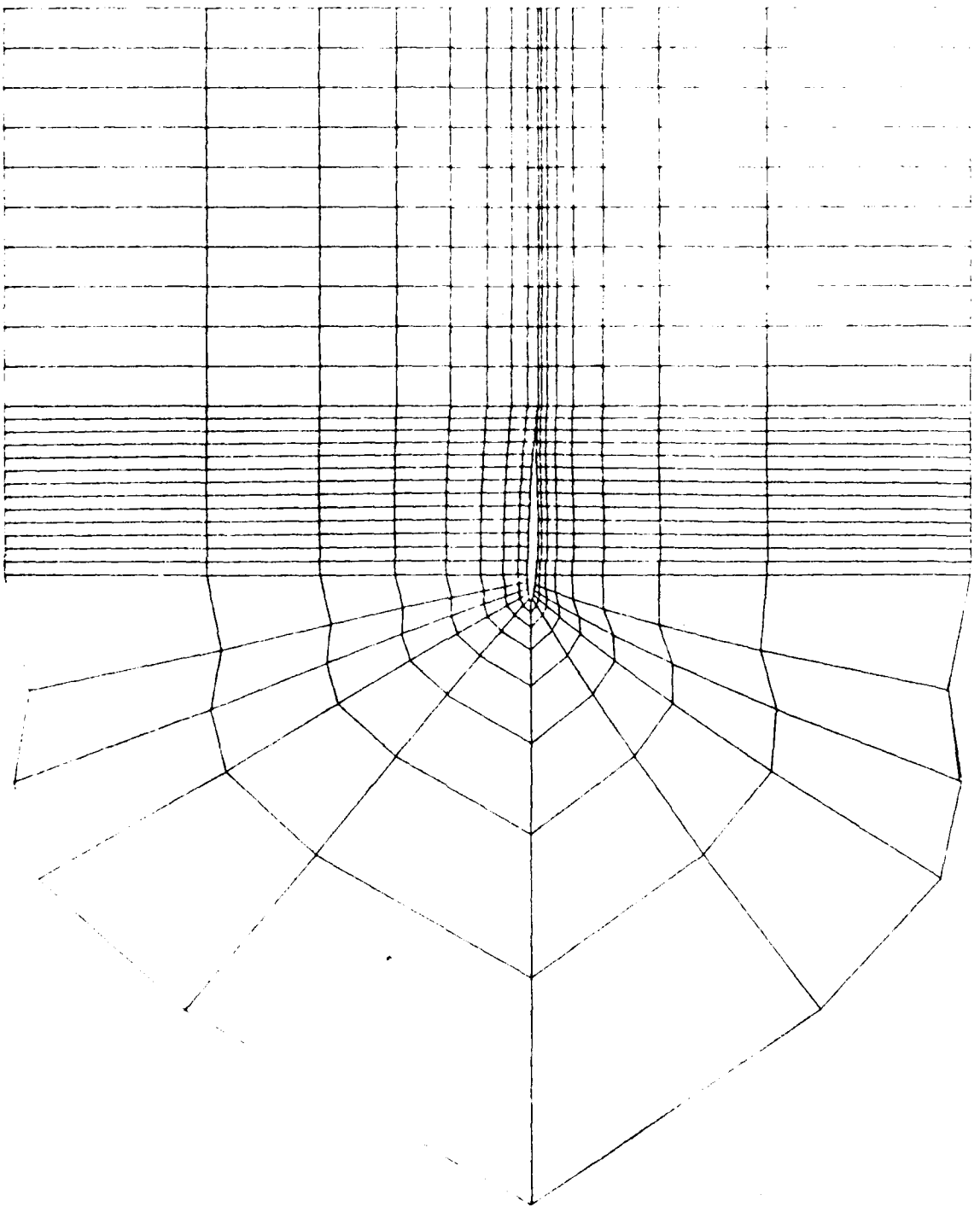


Fig. 4.10 Grid distribution, I-K section at the root

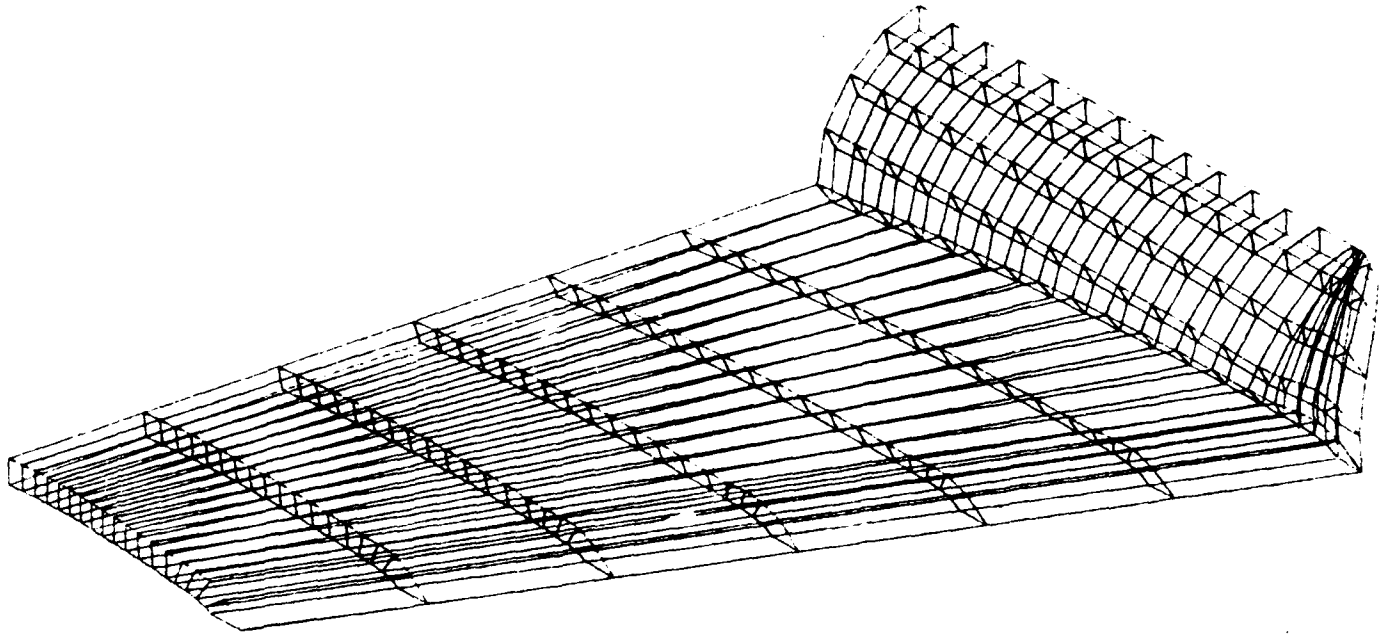


Fig. 4.11 The first layer of elements over the upper surface of wing-body combination

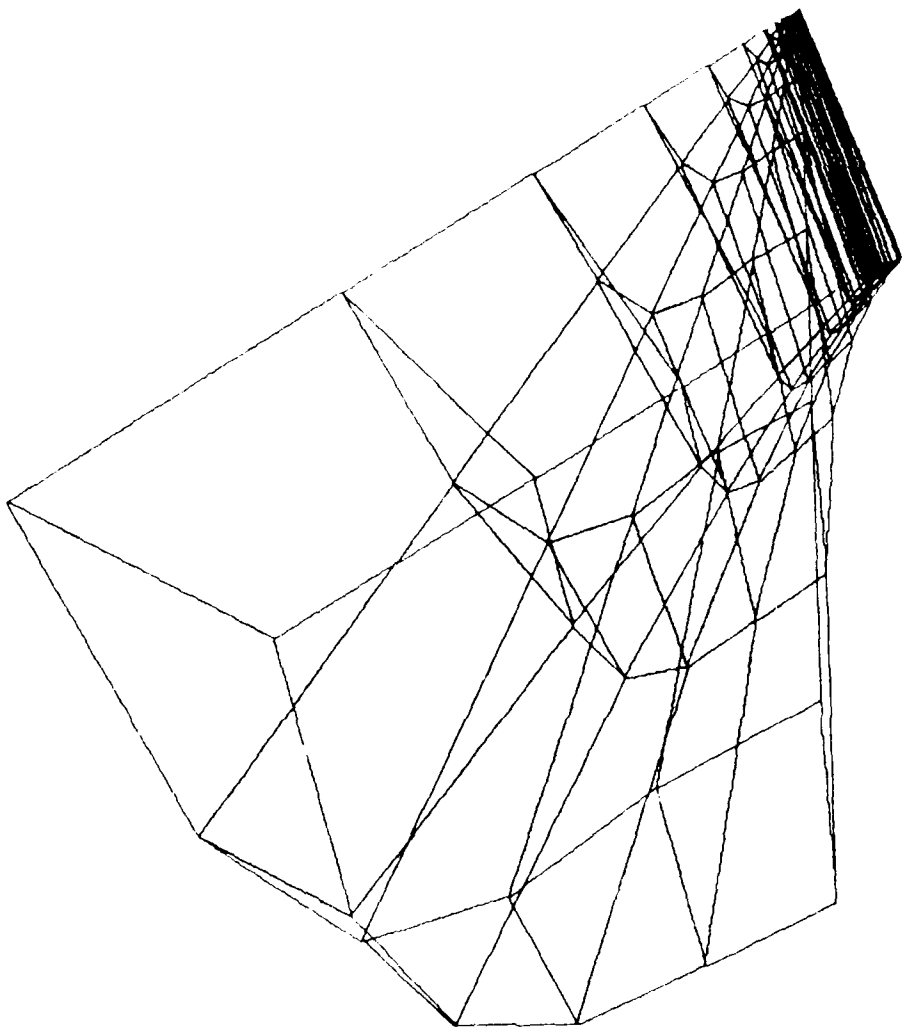


Fig. 4.12 The element distribution at the leading edge block over the wing only

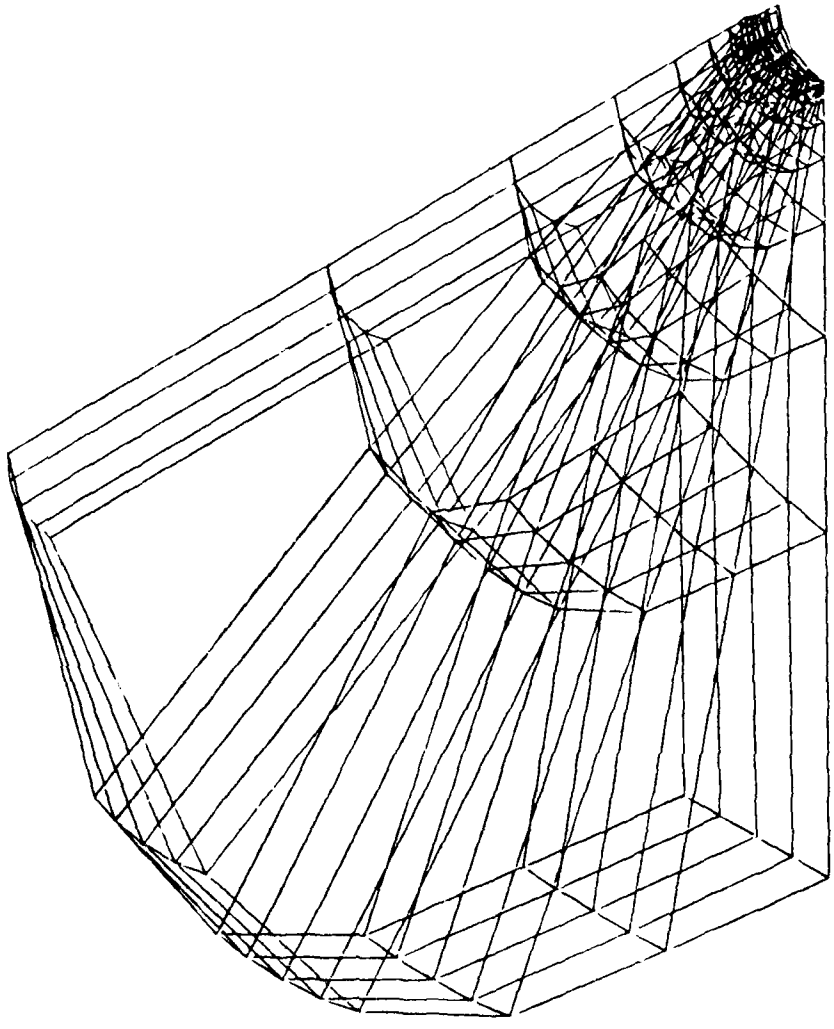


Fig. 4.13 The element distribution at the leading edge block over the body only

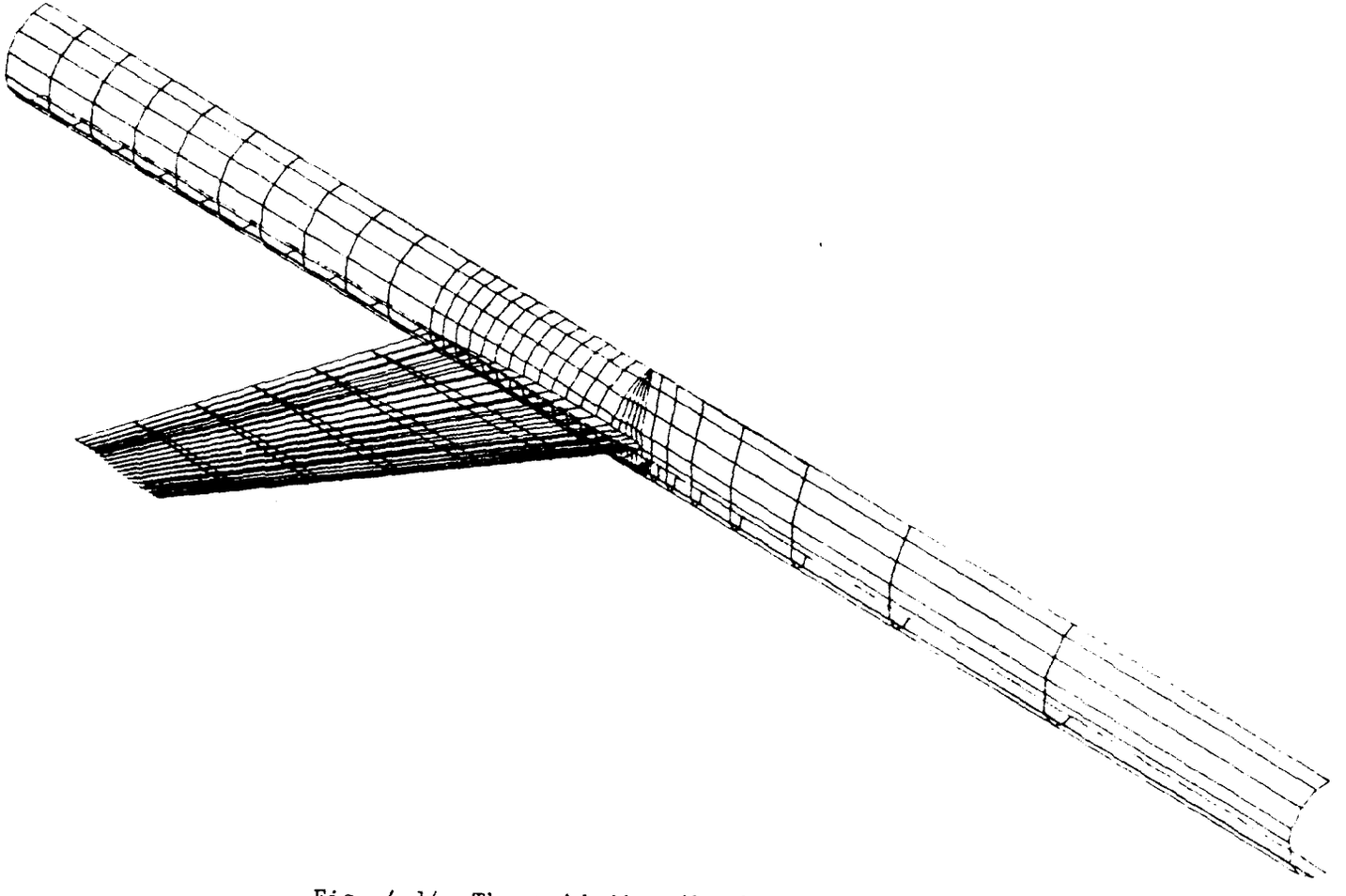


Fig. 4.14 The grid distribution over the whole wing-body combination

The flow characteristics for this problem are:

Free stream Mach number, $M_\infty = 0.9$

Inlet flow angle, $\beta_{in} = 3.0^o$

Outlet flow angle, $\beta_{out} = 3.0^o$

The initial solution was taken to be free-stream flow everywhere in the domain and a relatively high value of artificial viscosity was introduced. It was observed that, during the initial iterations, for the elements along which the Kutta-condition is applied, high velocity values were calculated. This was due to the sudden introduction of circulation. In order to eliminate these unattainable velocity values, for the elements above the vortex sheet, density values were assigned to be equal to the ones below the sheet, replacing the calculated densities. After the solution started to converge, exact density calculations were introduced to these elements with a linearly increasing percentage. After a reasonable convergence was obtained following the above scheme, the artificial voscosity was reduced. The iterations were continued until a convergent solution with a minimum amount of artificial viscosity was obtained.

The result of the flow analysis is expressed in terms of the pressure coefficient C_p. (Eq. 4.3.1) which is defined by non-dimensionalizing the pressure with respect to properties of the free stream.

$$C_p = \frac{P - P_\infty}{\frac{1}{2}P_\infty q_\infty^2} \tag{4.3.1}$$

For a perfect gas, pressure coefficient can be expressed in terms of free stream Mach number, M_∞, and the flow speed ratio; q/q_∞ [25] as follows:

$$Cp = \frac{2}{\gamma M_\infty^2} \{[1 + \frac{\gamma-1}{2} M_\infty^2 (1 - \frac{q^2}{M_\infty^2})]^{\gamma/(\gamma-1)} - 1\} \tag{4.3.2}$$

Local flow speed q can be obtained in terms of velocity potential;

$$q^2 = (\phi,_x^2 + \phi,_y^2 + \phi,_z^2) \tag{4.3.3}$$

Then, the equation 4.3.2 leads the following final definition of pressure coefficient in terms of variable ϕ:

$$C_p = \frac{2}{\gamma M_\infty^2} \{[1 + \frac{\gamma-1}{2} M^2 (1 - \frac{(\phi_x^2 + \phi_y^2 + \phi_z^2)}{q_\infty^2})]^{\gamma/(\gamma-1)} - 1\} \tag{4.3.4}$$

The pressure coefficient distributions obtained at the end of iterations for Case 1 are shown in Fig. 4.15a,b for the root and the tip sections of the wing and compared with experimental results [32]. The values shown in these figures are calculated at the centroids of the first row elements on the surface at these sections. The experimental results are the ones obtained on the surface and for the same flow conditions but for the isolated wing case.

When the results obtained from the first grid is compared with the experimental data. it is obvious that the solution around the leading edge is considerable smeared out. Furthermore, the Kutta-condition applied at the trailing edge does not seem to satisfy the equality of velocities at the upper and bottom edges of the trailing edge. Of course, it should be remembered that the centroidal values of the trailing edge elements have to be somewhat different. In order to improve the solution at these regions, it is practical to reduce element size so that a finer grid will be obtained and the centroids of the elements become closer to the surface. Beside these regions, it is observed that the solution is quite smooth over the flow domain except for the leading edge and the shock region. This suggests that one can employ a coarser grid over the wing past the leading edge and past the wing towards the far-field.

4.3.2 Case II. Modified Grid

Based on the above considerations, a second grid was designed in order to be able to show the effect of the distribution of the grid points and element densities, the total number of elements was kept the same. The new grid distribution along the spanwise direction while Fig. 4.16 shows the changes made along the streamwise direction. The block edges responsible for the leading edge were shortened and element numbers were increased as the orthogonality of the leading edge elements were preserved. The new block structure satisfying the above requirements is shown in Fig. 4.17. As it can be seen from this figure, upstream boundary is specified by two block edges rather than one, resulting in a better element distribution at the upstream. The curved radial edges of the blocks in surface-normal direction are to improve the orthogonality of the elements generated around the leading edge. The rest of the structure is quite similar to the previous one. Inside the

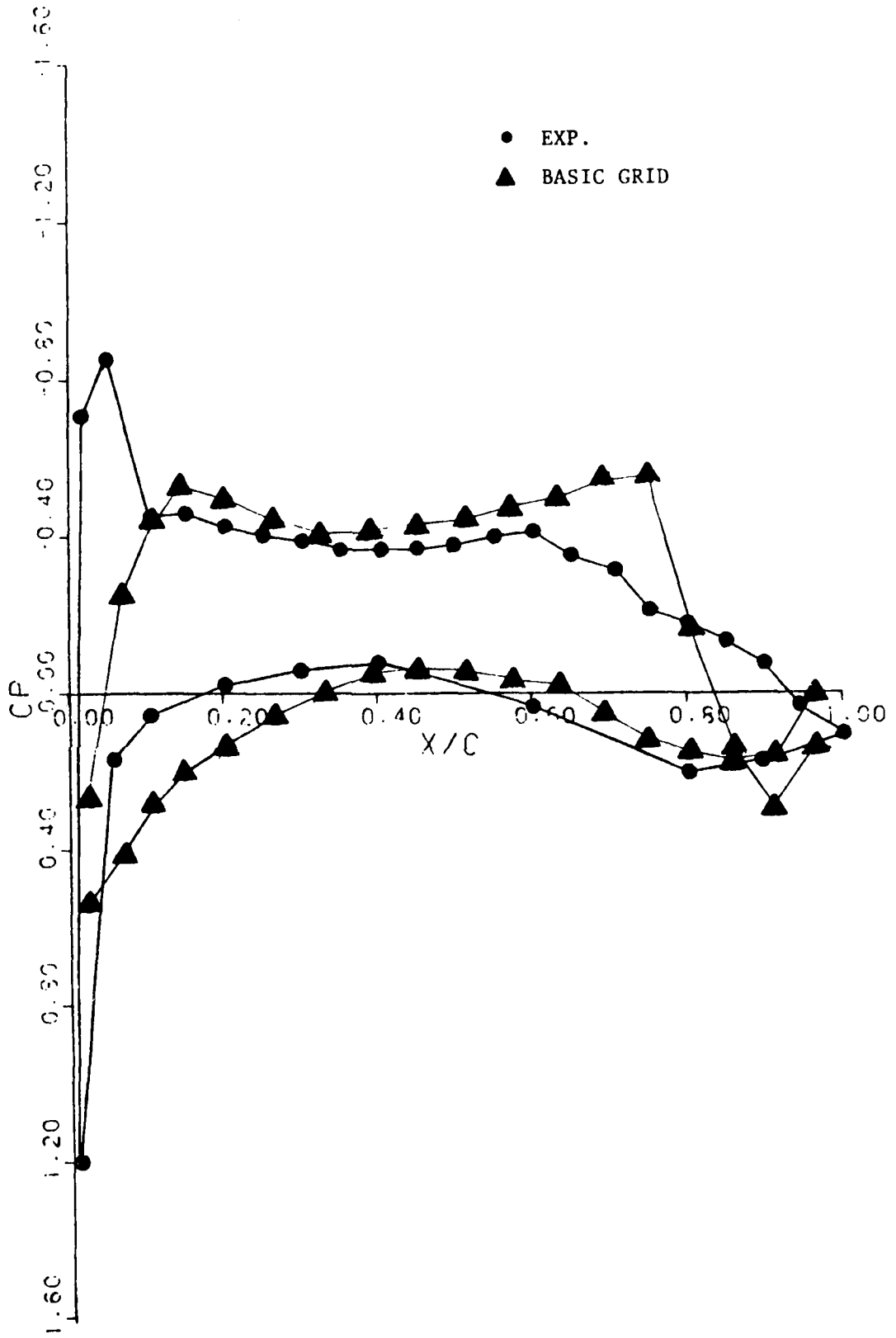


Fig. 4.15a Pressure distribution at the root

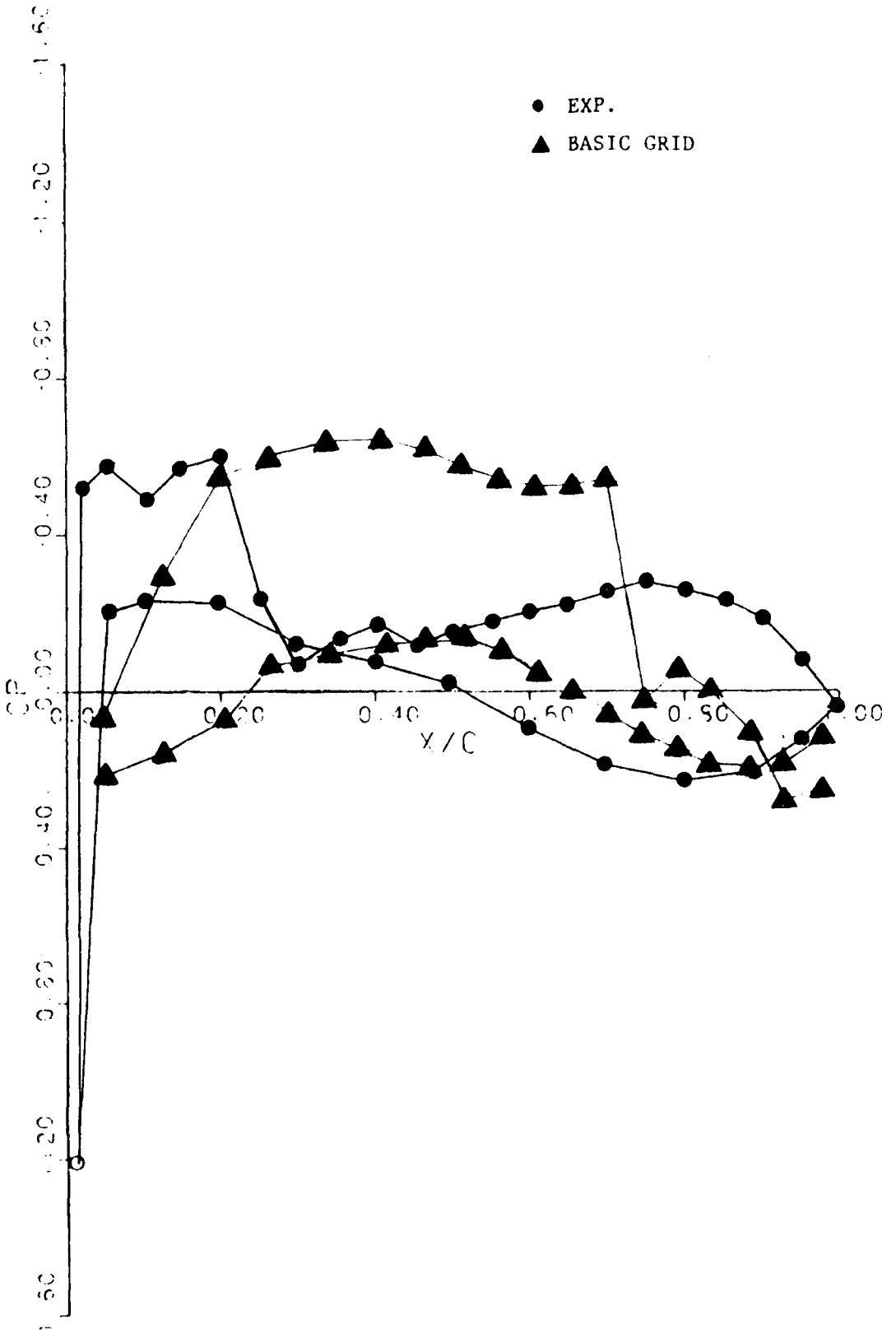


Fig. 4.15b Pressure distribution at the tip

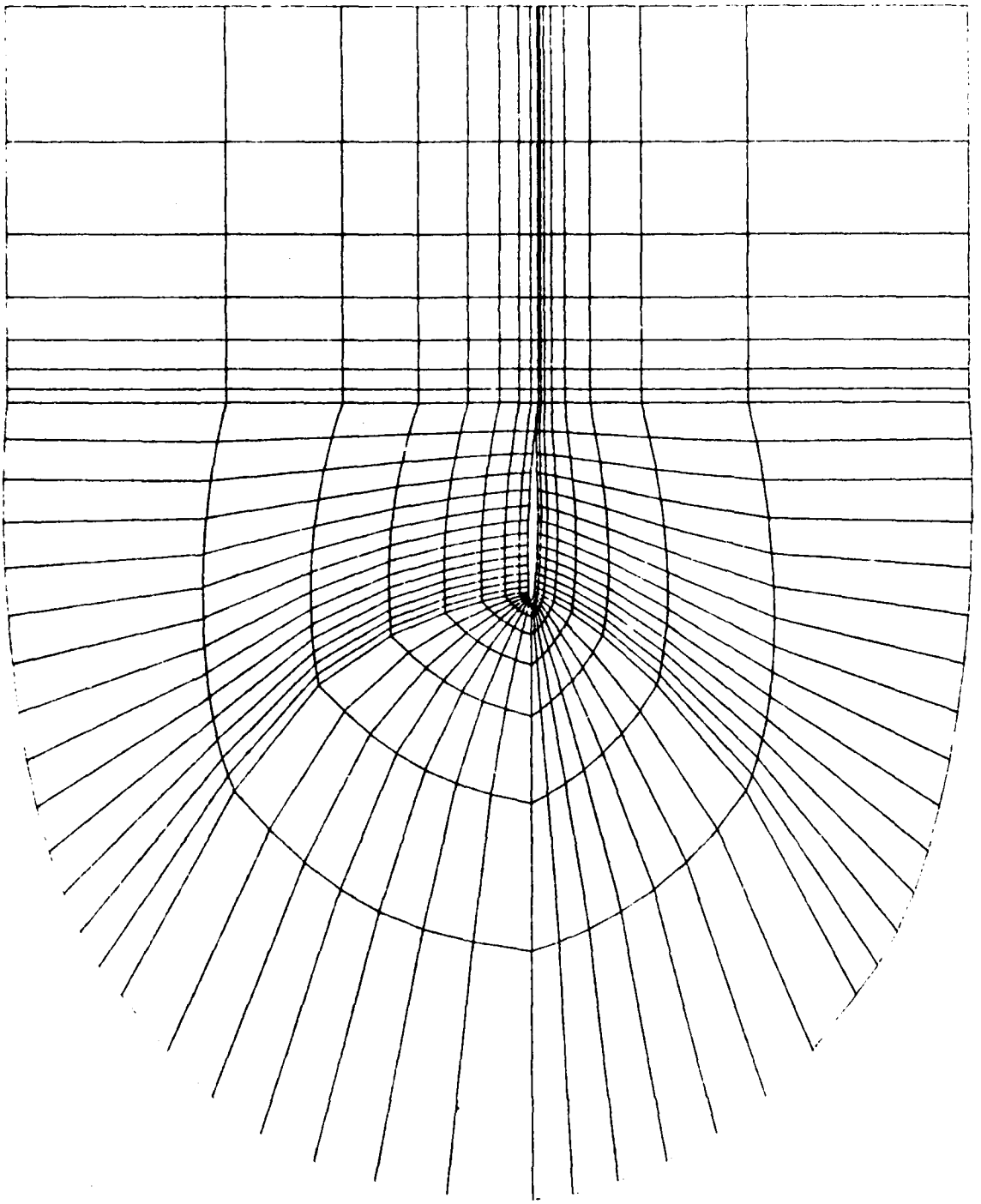


Fig. 4.16 The modified grid distribution, y-z section at the root

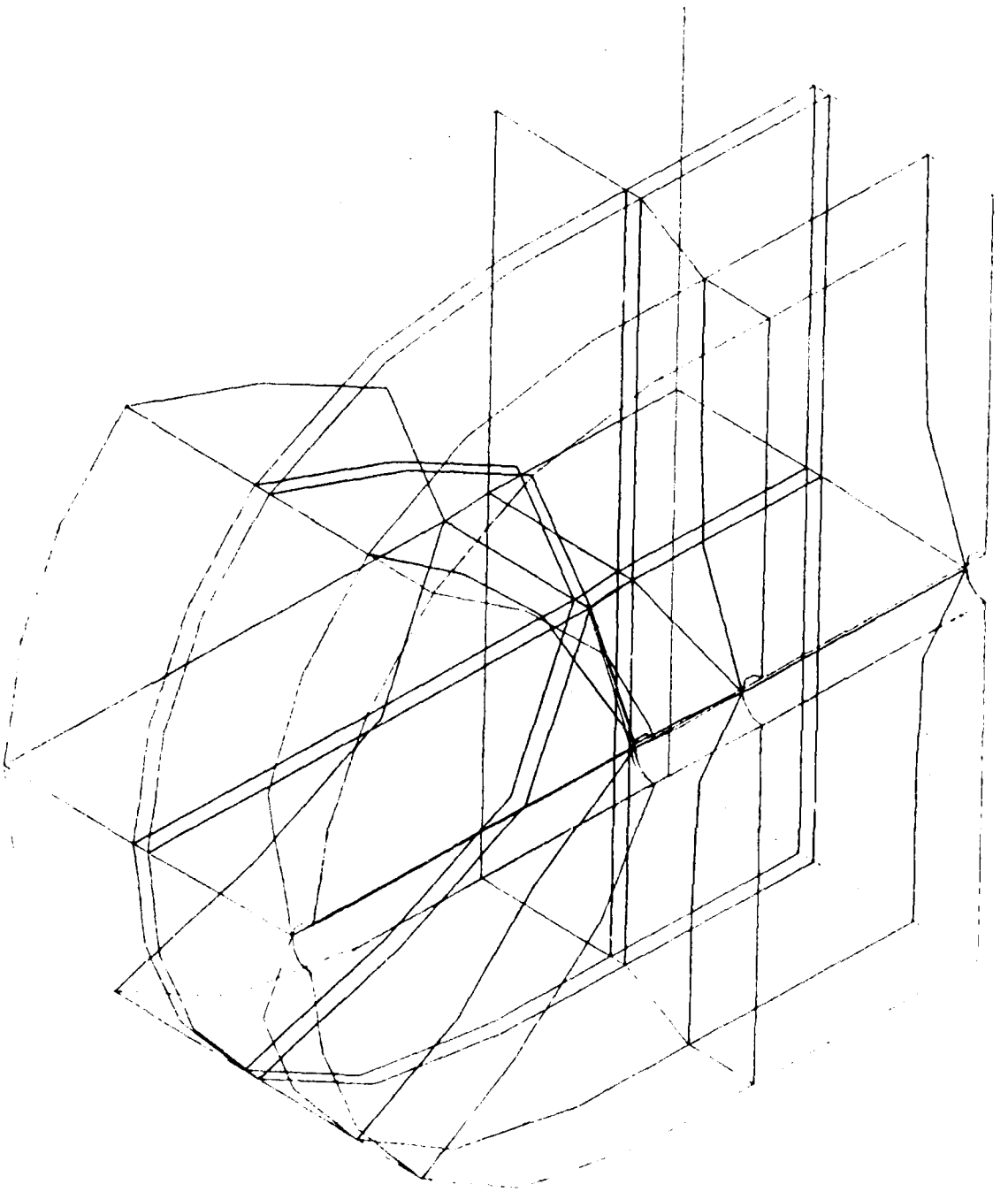


Fig. 4.17 The block structure for the modified grid

blocks, the element gradients were, on the other hand, modified to provide a finer element distribution towards the leading edge. At the far-field, the element number was reduced. However, by using different gradients smaller Kutta elements were placed around the trailing edge. The higher-order element specification was kept the same, while their sizes were modified.

Figure 4.18 shows the grid distribution over the wing-body and side boundary. Fig. 19a,b shows the location of higher-order elements for both grids. The second modified grid was employed for the solution of the same flow problem. The pressure distributions obtained from the solution are presented in Fig. 4.20a-b at several sections along the wing span.

Both test cases were ran on an IBM 4341 system. Approximate CPU time for a single iteration was around 20 minutes. Considerable amount of this time is spent in the file manipulations of the constant coefficient matrix. Due to the limited number of tape drives, this procedure required back-spacing of a tape which is rather inefficient.

The first case converged after 260 iterations and the results were obtained with the artificial viscosity content of $\mu = 2$. The employed convergence criteria was the maximum normalized change in the calculated Mach numbers of the elements in the whole domain which was in the order of 10^{-4} for the last iteration.

In the second case convergent solution with artificial viscosity content of $\mu = 1.5$, was obtained at the 325th iteration.

It should be kept in mind that the experimental results belong to the wing-alone case in which the pressure coefficient at the leading edge has a tendency to be higher than that of the wing-body combination case.

4.3.3 Comparison of Results

The inspection of results obtained for two different grids with the same number of elements, reveals some important aspects of the design of computational grids. They show that the grid distribution which provides a better alignment with the changing flow variables also produces a better agreement with experimental results. The element densities specified at the regions of high flow gradients, to a great extent, may become responsible for the accurate solution in the whole domain in transonic flows.

Results in fig. 4.20 show that the second grid agrees better with experiments as far as leading edge singularity and Kutta-condition are concerned. Figures 4.19a,b show the element distribution around the wing

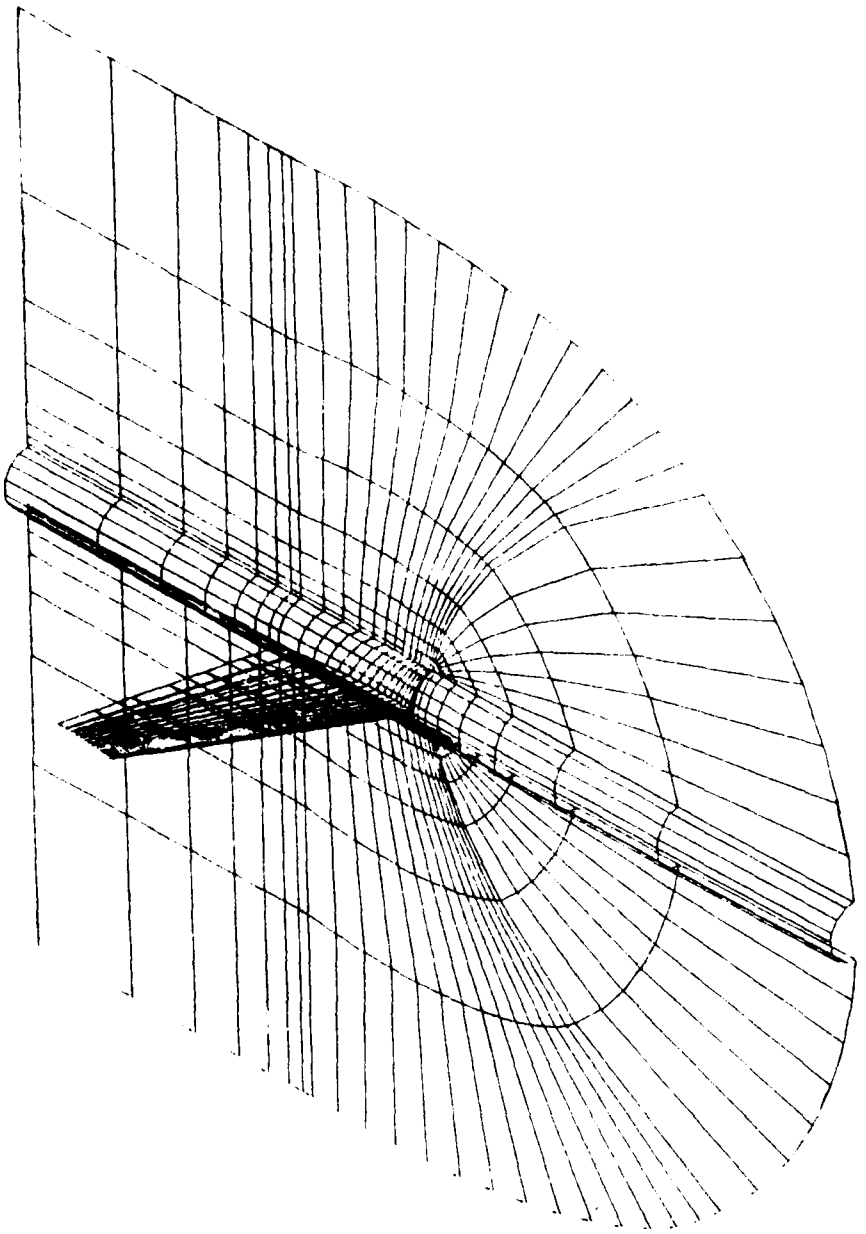


Fig. 4.18 Grid distribution over the wing-body and side boundary

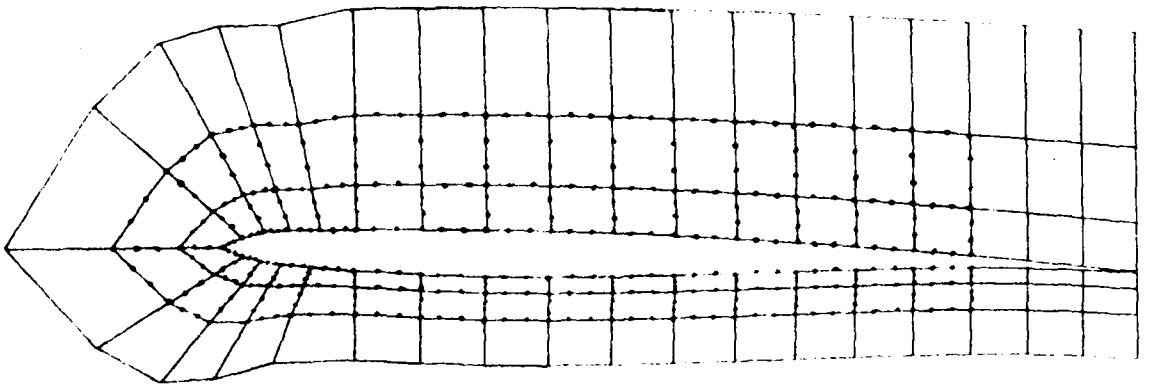


Fig. 4.19a The first grid distribution around wing-root section and higher-order elements

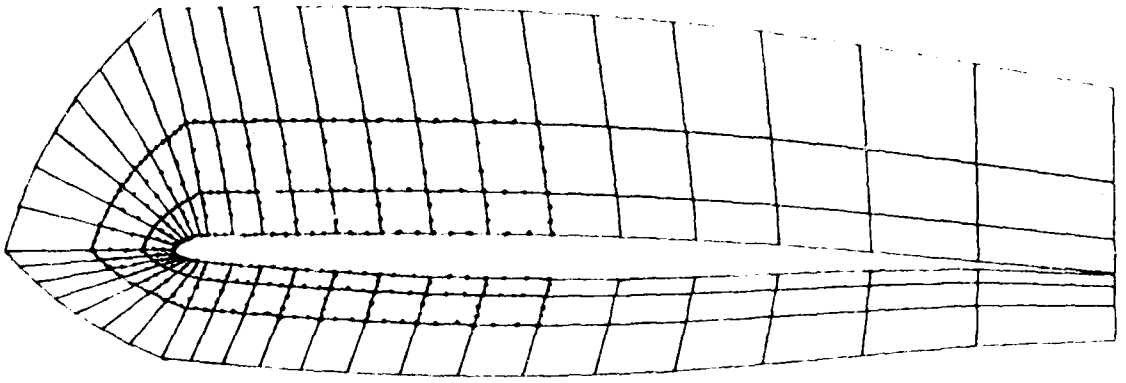


Fig. 4.19b The modified grid distribution around wing-root section and higher-order elements

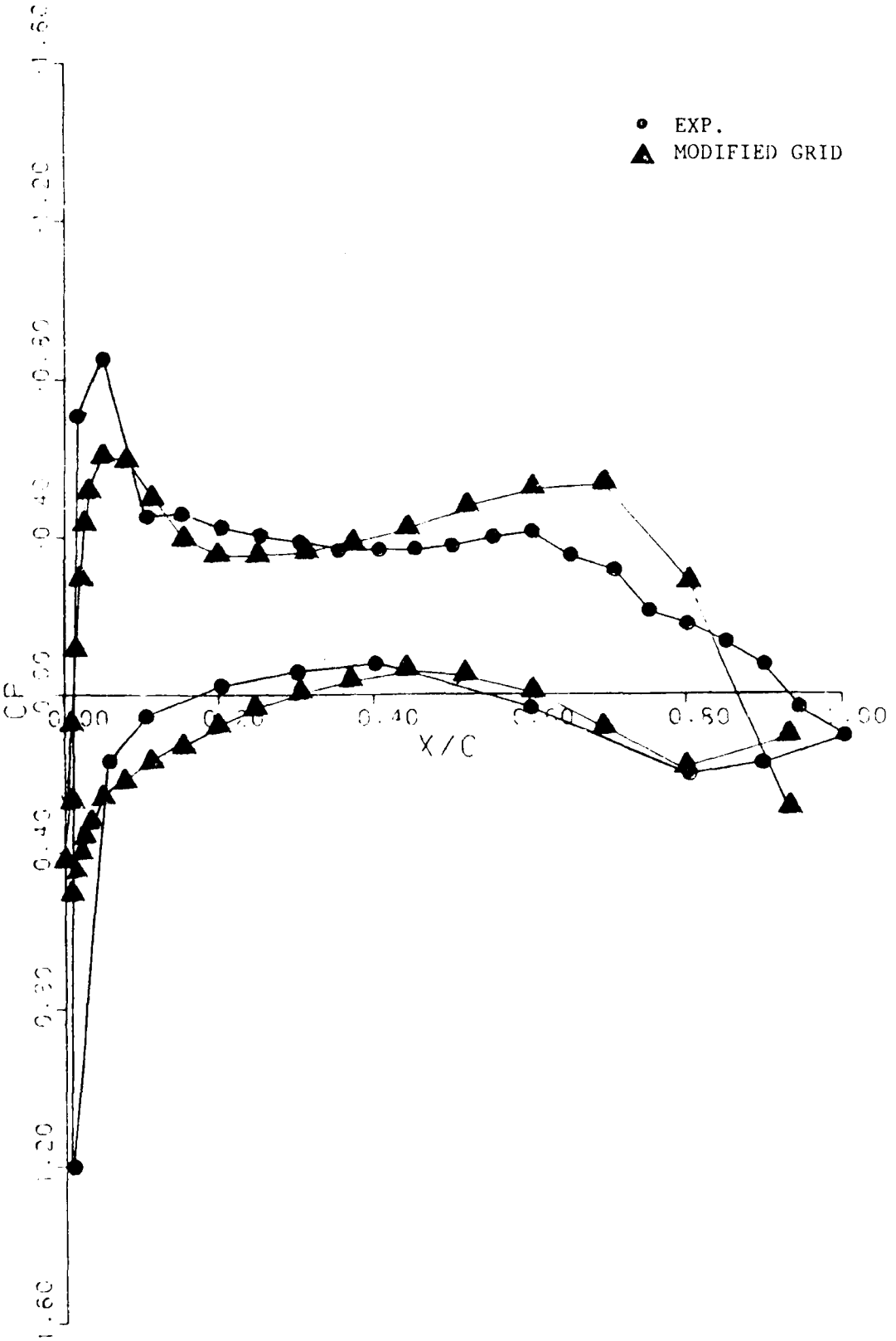


Fig. 4.20a Pressure distribution at the root section of the modified grid (22% of the wing span)

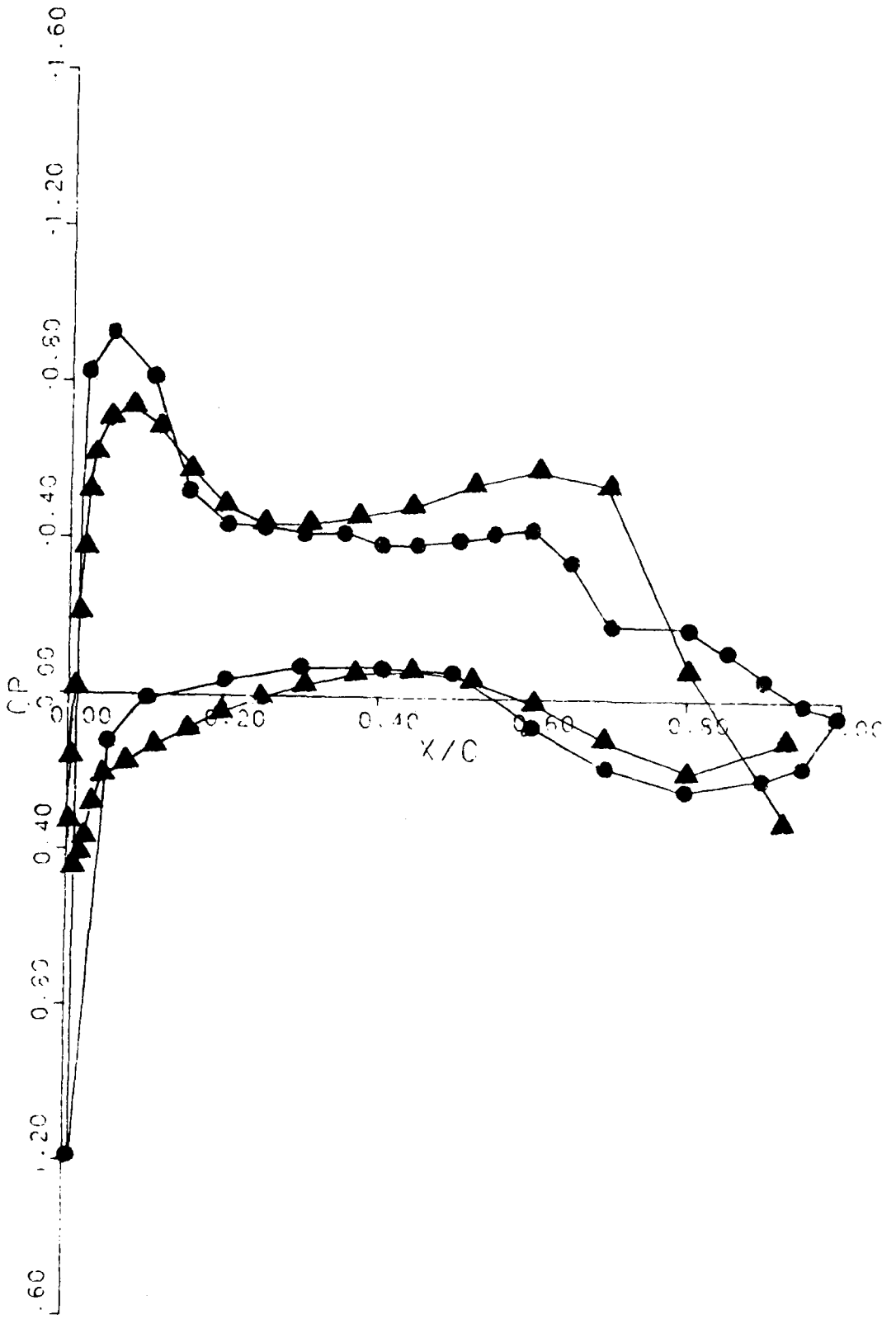
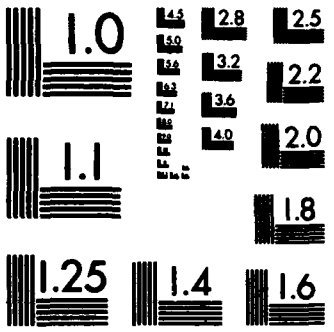


Fig. 4.20b Pressure distribution at 40% of the wing span away from the root.

AD-A135 744 ANALYSIS OF THREE-DIMENSIONAL TRANSONIC POTENTIAL FLOWS USING OPTIMUM GRID(U) INDIANA UNIV-PURDUE UNIV AT INDIANAPOLIS SCHOOL OF ENGINEERIN.. A ECER DEC 82 2/2

UNCLASSIFIED AFOSR-TR-83-1052 AFOSR-80-0258 F/G 20/4 NL

END
1
FILMED
1-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

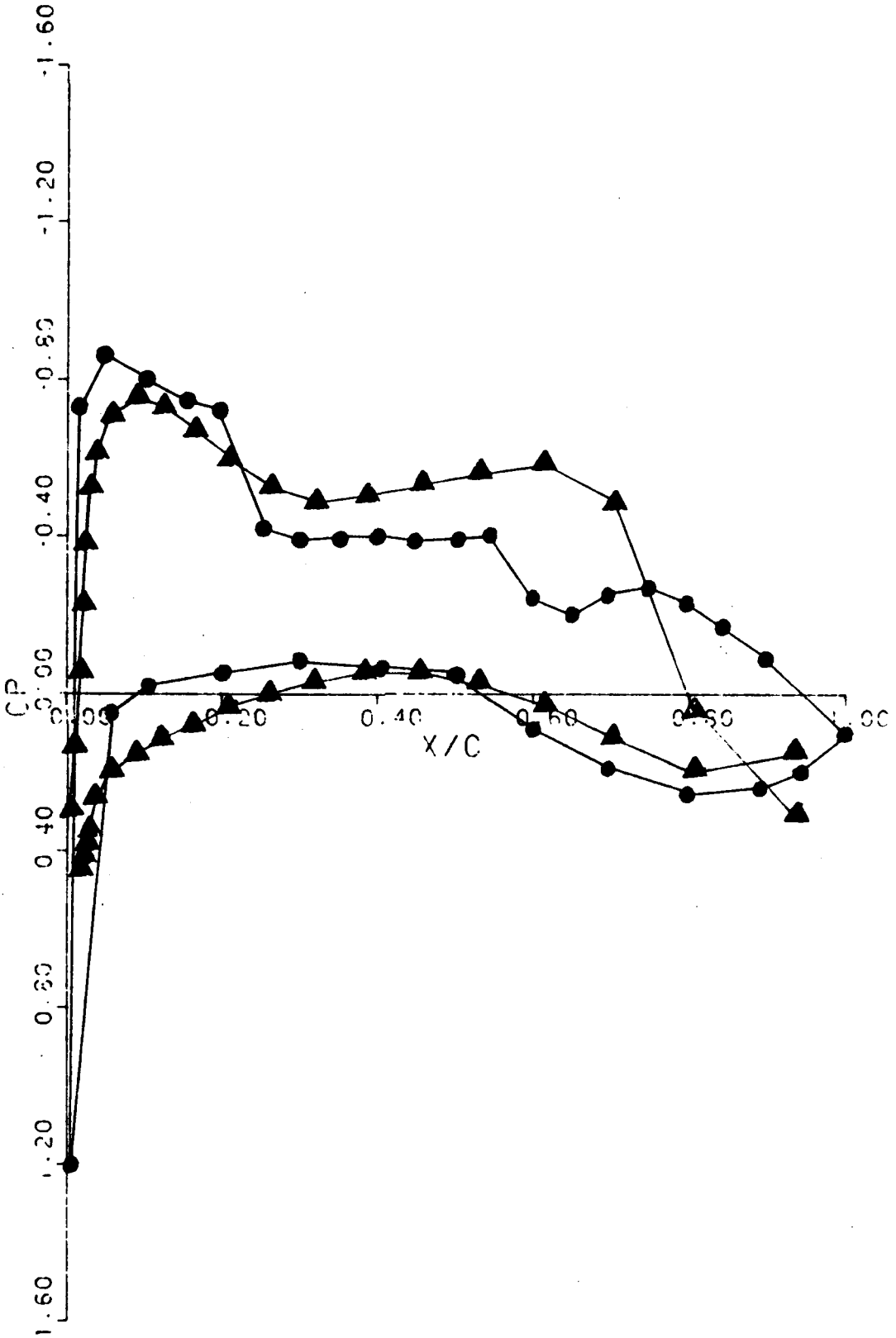


Fig. 4.20c Pressure distribution at 60% of the wing span away from the root.

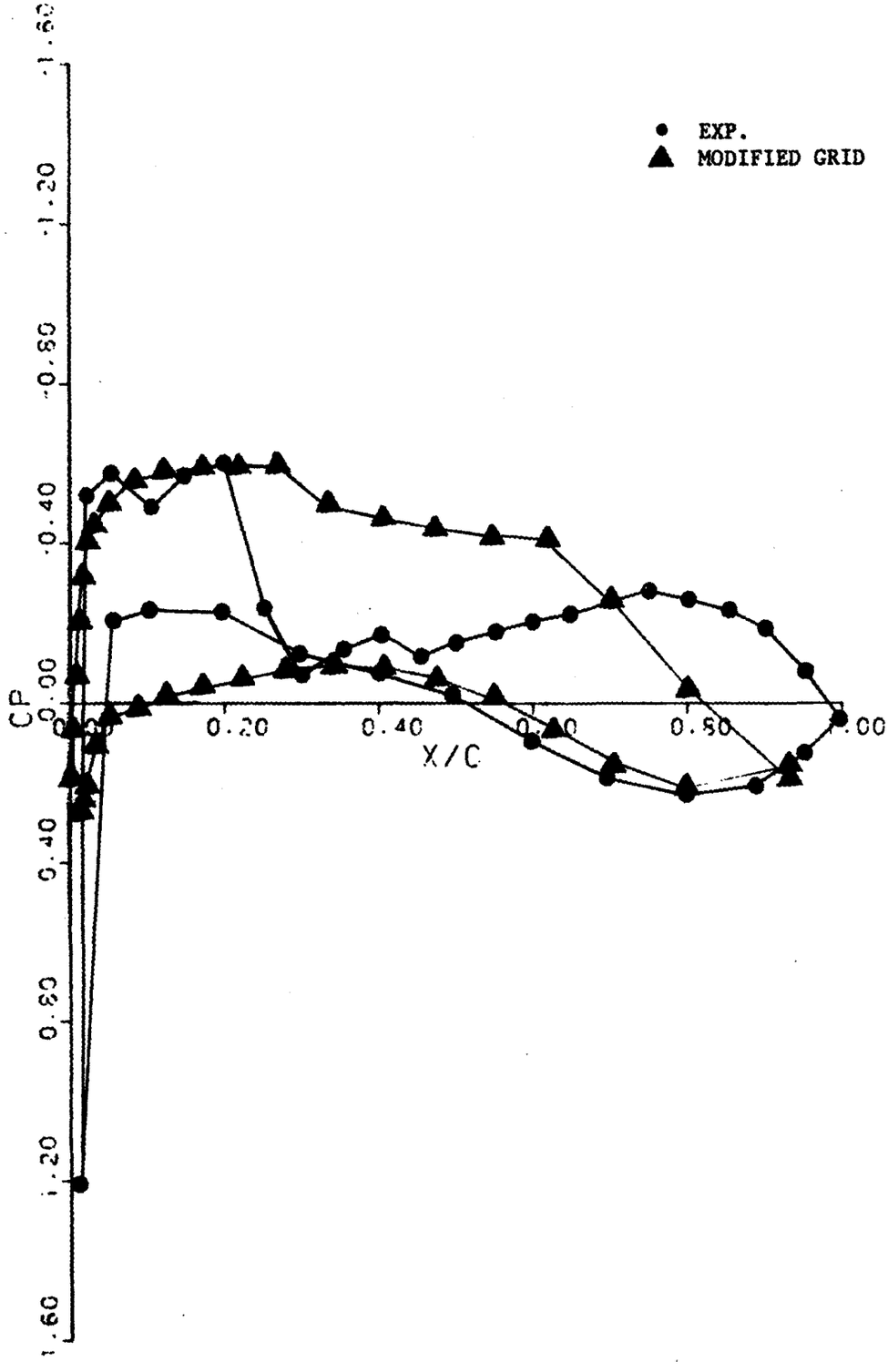


Fig. 4.20d Pressure distribution at the tip section of the modified grid (95% of the wing span)

for each of the grids and the location of higher-order elements. As it can be seen from figure 4.19a, density of elements around the leading edge and the reduction in the distortion of these elements seem to be responsible for the improved solution at the leading edge singularity.

The improved solution at the trailing edge is attributed to the modifications done at this region resulting in smaller elements. Looking at the almost identical lower surface pressure distributions except at the leading edge, it is concluded that the lower surface remains subsonic and is not sensitive to inaccuracies at the leading edge.

Although the modified grid gives better and improved results at the root section of the wing, it was noticed that the shock at the tip section moved further downstream. This was due to the increase in the element size at this region as a result of biasing the same number of elements towards the leading edge. These results show the necessity of providing sufficient grid refinement for all flow regions around the wing. On the other hand, the inefficiency of using equally-spaced grid points is also apparent. The ratio of spacing between the nodes around the leading edge to the nodes in larger elements is about 1/80.

The application of the finite element method for generating a block-structured computational grid provides the necessary flexibility for designing the type of grids discussed above. For solving a practical three-dimensional problem, one can start with a basic grid and obtain some preliminary results. This grid can then be refined for better accuracy. The comparison of results from the basic and modified grids will indicate the accuracy of the obtained solution even without the aid of experimental results. In the developed finite element scheme, one can concentrate on a particular block of the basic grid and modify it with a minimum effort without chancing the remaining portions of the computational grid.

Although the present application demonstrates a user oriented adaptive grid generation, a mathematically described automative grid generation scheme can easily be implemented to the present grid generation scheme.

4.4 CONCLUDING REMARKS

The main objective of the study was to evaluate the concept of an "optimum grid" for analyzing three-dimensional transonic flows. For this purpose, we studied the importance of computational grids in terms of both the accuracy and efficiency of a computational procedure. For the sample problem we have chosen, it was necessary to use over 105 grid points to analyze an isolated wing with existing finite difference wing codes. On the other hand, if a certain flow region is not modeled accurately, the errors can corrupt the entire solution in transonic flows. The use of "artificial viscosity" requires extreme care in the analysis of transonic flows and is closely related to the computational grid. Based on these requirements, we concluded that a computational grid generation scheme needs to be very flexible in dealing with such problems. The following considerations summarize our development in this area:

a) The grid generation scheme presented in this paper is based on a block structure and on automatic mesh generation for each block. This allows the flexibility in designing different types of grids for different flow regions. One has to consider the design of the grid for a single block at a time. The grid points between the blocks are matched automatically. It is also flexible in a sense that if the grid for a certain flow region has to be refined, again only that particular block is modified at that time.

b) The design of efficient and accurate grid requires an understanding of the flow field to be analyzed. Only then, one can design a grid close to an "optimum grid." One has to design an adaptive process for this purpose. In the case, where no prior knowledge of the flow field exists, one can start with a basic grid, obtain some numerical results and then adapt the grid to suit the flow field. Leading edge, trailing edge, shock, etc., are all critical areas where one has to experiment for designing an optimum grid. In the present procedure, one can choose a particular critical flow region, and convert a group of elements from linear to cubic in that particular region. In the present study, this was illustrated for the leading edge problem. Consequently, the number of grid points in that particular area is increased

considerably. Similarly, one can increase the number of elements in each block or change the distribution of grid spacing in one block. All these operations require minimal input. With these capabilities, one can work with grids where the number of grid points does not reach 10^6 for three-dimensional flow problems.

c) Finite element method brings considerable freedom in modeling complex geometries. Relocation of boundary nodes on a surface improves the accuracy of boundary conditions. As mentioned above, the use of different order finite elements in the same grid enables better modeling of curved surfaces.

d) One can develop a separate mesh generation procedure for each of the blocks. One then can have, for example, a leading edge block or a shock block for which a optimum mesh generation procedure is developed previously.

e) In the present procedure, the blocks are still assembled together at the end as a single computational grid and analyzed as a single block of grid points. This requires matching of grid points at block interfaces, which is automatically performed with the present scheme. Further development of this present concept would be to remove this restriction where each block can be meshed independently. This will provide further flexibility in designing optimum grids for each of the local flow regions, without globally increasing the number of node points.

REFERENCES

1. Murman, E. M. and Cole, J. D., "Calculation of Plane Steady, Transonic Flows", AIAA J., Vol. 20, p. 114, 1971.

2. Ballhaus, W. F. and Bailey, F. R., "Numerical Calculations of Transonic Flow about Swept Wings", AIAA Paper 72-677, 5th Fluid and Plasma Dynamics Conference, Boston, Mass., June 26-28, 1972.

3. Bailey, F. R. and Ballhaus, W. F., "Relaxation Methods for Transonic Flow about Wing-Cylinder Combinations and Lifting Swept Wings", The Third International Conference on Numerical Methods in Fluid Dynamics, Paris, France, July 3-7, 1972.

4. Jameson, A., "Iterative Solution of Transonic Flows over Airfoils and Wings including Flows at Mach 1", Comm. on Pure and Applied Math., Vol. 27, p. 283-304, 1974.

5. Bailey, F. R. and Ballhaus, W. F., "Comparisons of Computed and Experimental Pressures for Transonic Flows about Isolated Wings and Wing-Fuselage Configurations", N76-10051.

6. Ballhaus, W. F., Jameson, A., Albert, J., "Implicit Approximate Factorization Schemes for Steady Transonic Flow Problems", AIAA J., Vol. 16, No. 6, June 1978.

7. Hafez, M., South, S., Murman, E. M., "Artificial Compressibility Methods for Numerical Solutions of Transonic Full Potential Equation", AIAA J., Vol. 17, No. 8, August 1979.

8. Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods", AIAA J., Vol. 17, No. 5, May 1974.

9. Chen, L. T., "A More Accurate Transonic Computational Method for Wing-Body Configurations", AIAA 20th Aerospace Sciences Meeting, Orlando, Florida, January 11-14, 1982.

10. Eberle, A., "Finite Element Method for the Solution of Full Potential Equations in Transonic Steady and Unsteady Flow", Theoretical Aerodynamics FE 122, 1980.

11. Deconinck, H. and Hirsh, Ch., "Finite Element Methods for Transonic Blade to Blade Calculation in Turbo-Machines", ASME 26th International Gas Turbine Conference, Houston, TX., March 1981.

12. Akay, H. U. and Ecer, A., "Treatment of Shocks in the Computation of Transonic Flows using Finite Elements", Proceedings of the Third International Conference on Finite Elements in Flow Problems, Vol. 1, p. 341-344, Banff, Alberta, Canada, June 10-13, 1980.

13. Ecer, A. and Akay, H. U., "Finite Element Analysis of Transonic Flows in Cascades-Importance of Computational Grids in Improving Accuracy and Convergence", NASA Contractor Report 3446, July 1981.

14. Ecer, A., Akay, H. U., Bhutta, B. A., "Finite Element Computations of Three-Dimensional Transonic Flow". ASME Symposium on Computers in Flow Predictions and Fluid Dynamics Experiments, Washington, D.C., November 1981.

15. Yurtseven, H. O., Karaca, C. and Ecer, A., "Fast, Pseudo-time Integration Scheme for the Solution of Steady Transonic Flow Problems", Proceedings of 2nd International Conference of Finite Elements in Non-linear Mechanics, Stuttgart, Germany, August 25-28, 1981.

16. Ecer, A., Citipitioglu, E. and Bhutta, B. A., "Design of Finite Element Grids for the Computation of the Three-Dimensional Transonic Flow around a Wing", AIAA/ASME - 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, June 7-11, 1982.

17. Thompson, J. F., Thames, F. C. and Mostun, C. S., "Boundary-Fitted Curvilinear Coordinate Systems for Solution of Partial Differential Equations on Fields Containing any Number of Arbitrary Two-Dimensional Bodies", J. of Computational Physics, Vol. 15, p. 294-314, 1974.

18. Lee, K. D., "3-D Transonic Flow Computations using Grid Systems with Block Structure", Computational Fluid Dynamics Conference, June 22-23, 1981.

19. Chen, L. T. and Caughey, D. A., "A Nearly Conformal Grid Generation Method for Transonic Wing-Body Flowfield Calculations", AIAA 20th Aerospace Sciences Meeting, AIAA 82-0108, Orlando, Florida, January 11-14, 1982.

20. Yu, N. O., "Grid Generation and Transonic Flow Calculation for Three-Dimensional Configurations", AIAA 13th Fluid and Plasma Dynamics Conference, No. AIAA 80-1391, Snowmass, Colorado, July 14-16, 1980.

21. Rizzi, A. and Erikson, L. E., "Transfinite Mesh Generation and Dumped Euler Equation Algorithm for Transonic Flow around Wing-Body Configurations", AIAA Computational Fluid Dynamics Conference, No. AIAA 81-0999, June 22-23, 1981.

22. Holst, T. L. and Brown, D., "Transonic Airfoil Calculations using Solution-Adaptive Grids", AIAA Computational Fluid Dynamics Conference, No. AIAA 81-0101, June 23-23, 1981.

23. Dwyer, H. A., Kee, R. J. and Sanders, B. R., "Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer", AIAA J., Vol. 18, No. 10, October 1980.

24. Deconinck, H. and Hirsh, Ch., "Finite Elements Methods for Transonic Blade to Blade Calculations in Turbo-Machines", ASME 26th International Gas Turbine Conference, Houston, TX, March 1981.

25. Zucrow, M. J. and Hoffman, J. D., "Gas Dynamics", Vol. I., John Wiley and Sons, Inc., 1976.

26. Bristau, M. O., Peronneau, O., Glowinsky, R., Periqus, J., and Perrier, P., "On the Numerical Solution of Non-Linear Problems in Fluid Dynamics by Least Square Method and Finite Element Methods. (I) Least Square Method and Conjugate Gradient Solution of the Continuous Problem", Comp. Appl. Mech. Engg., 17/18, 1979.

27. Steger, J. L. and Baldwin, B. S., "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow", NASA Technical Note, NASA TN 0-6997, October 1972.

28. Ecer, A. and Akay, H. U., "Investigation of Transonic Flow in a Cascade using an Adaptive Mesh", AIAA 13th Fluid and Plasma Dynamics Conference, AISS 80-1430, Snowmass, Colorado, July 14-16, 1980.

29. Lomax, H., Kutter, P. and Fuller, F. B., "The Numerical Solutions of Partial Differential Equations Governing Convection", AGARD ograph No. 146.

30. Becker, E. B., Carey, G. E., Oden, J. T., "Finite Elements, an Introduction", Vol. 1, Prentice-Hall Inc., 1981.

31. Conte, S. D. and Deboor, C., "Elementary Numerical Analysis", McGraw-Hill Inc., 1972.

32. Hinson, B. L and Burdges, K. P., "Acquisition and Application of Transonic Wing and Far-Field Test Data for Three-Dimensional Computational Method Evaluation", Final Report, Lockheed, Georgia Company, AFOSR-TR 80-0421, May 1978 - August 1979.

33. Citipitioglu, E., "Universal Serendipity Elements". International Journal of Numerical Methods in Engineering, to be published in 1983.

34. Citipitioglu, E., Spyropoulos, J., Tuncer, I. H., "Isomap, Three-Dimensional Finite Element Grid Generation Program, User's Manual", Division of Engineering, IUPUI, 1983.

APPENDIX-A

GAUSSIAN INTEGRATION

Gaussian integration scheme can be employed to determine exact integration of polynomials over regular shapes e.g. rectangular blocks, etc. This scheme has become an important part of the finite element technique and is commonly used. The integration is performed by using a formula which is expressed in terms of summation of values of the integrand at certain 'Gauss' points, P_i, multiplied by corresponding weights, W_i.

As it is pointed out in section 2.3.3, the degree of accuracy in the integration formula is very important in terms of the accuracy of the results and the computer time spent on the numerical integration. In the present study, although, maximum three points integration formula is employed, the following table A.1 supplies the integration points up to ten points and the corresponding weights.

Table A.1 Integration points and weights for the Gauss-Legendre quadrature formula

P_i	W_i
$n = 1$	
0	2 00000 00000 00000
$n = 2$	
0 57735 02691 89626	1 00000 00000 00000
$n = 3$	
0 77459 66692 41483	0 55555 55555 55556
0 00000 00000 00000	0 88888 88888 88889
$n = 4$	
0 86113 63115 94053	0 34785 48451 37454
0 33998 10435 84856	0 65214 51548 62546
$n = 5$	
0 90617 98459 38664	0 23692 68850 56189
0 53846 93101 05683	0 47862 86704 99366
0 00000 00000 00000	0 56888 88888 88889
$n = 6$	
0 93246 95142 03152	0 17132 44923 79170
0 66120 93864 66265	0 36076 15730 48139
0 23861 91860 83197	0 46791 39345 72691
$n = 7$	
0 94910 79123 42759	0 12948 49661 68870
0 74153 11855 99394	0 27970 53914 89277
0 40584 51513 77397	0 38183 00505 05119
0 00000 00000 00000	0 41795 91836 73469
$n = 8$	
0 96028 98564 97536	0 10122 85362 90376
0 79666 64774 13627	0 22238 10344 53374
0 52553 24099 16329	0 31370 66458 77887
0 18343 46424 95650	0 36268 37833 78362
$n = 9$	
0 96816 02395 07626	0 08127 43883 61574
0 83603 11073 26636	0 18064 81606 94857
0 61337 14327 00590	0 26061 06964 02935
0 32425 34234 03809	0 31234 70770 40003
0 00000 00000 00000	0 33023 93550 01260
$n = 10$	
0 97390 65285 17172	0 06667 13443 08688
0 86506 33666 88985	0 14945 13491 50581
0 67940 95682 99024	0 21908 63625 15982
0 43339 53941 29247	0 26926 67193 09996
0 14887 43389 81631	0 29552 42247 14753


FILM
1-8